

University of South Alabama

JagWorks@USA

Theses and Dissertations

Graduate School

5-2022

Circuit-Variant Moving Target Defense for Side-Channel Attacks on Reconfigurable Hardware

Tristen H. Mullins

University of South Alabama, tah1323@jagmail.southalabama.edu

Follow this and additional works at: https://jagworks.southalabama.edu/theses_diss



Part of the [Information Security Commons](#), and the [Other Computer Sciences Commons](#)

Recommended Citation

Mullins, Tristen H., "Circuit-Variant Moving Target Defense for Side-Channel Attacks on Reconfigurable Hardware" (2022). *Theses and Dissertations*. 54.

https://jagworks.southalabama.edu/theses_diss/54

This Dissertation is brought to you for free and open access by the Graduate School at JagWorks@USA. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of JagWorks@USA. For more information, please contact jherrmann@southalabama.edu.

CIRCUIT-VARIANT MOVING TARGET DEFENSE FOR SIDE-CHANNEL
ATTACKS ON RECONFIGURABLE HARDWARE

A Dissertation

Submitted to the Graduate Faculty of the
University of South Alabama
in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

in

Computing

by

Tristen H. Mullins

B. S., University of South Alabama, 2018

May 2022

To my husband Alex and our children, Charlotte and Beau Mullins.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my committee chair, Dr. Todd Andel for his valued guidance and encouragement throughout my time at South Alabama. I would also like to thank my committee members Dr. Todd McDonald, Dr. Dimitrios Damopoulos, and Dr. Samuel Russ for lending their expertise in the completion of this research.

The completion of this dissertation would not have been possible without the unwavering support of my family. I am extremely grateful to my husband and children for the inspiration they provided and sacrifices they made to allow me to pursue this degree.

TABLE OF CONTENTS

	Page
LIST OF TABLES.....	vi
LIST OF FIGURES	vii
LIST OF ABBREVIATIONS.....	ix
ABSTRACT.....	x
CHAPTER I INTRODUCTION.....	1
1.1 Motivation.....	1
1.2 Problem Statement.....	2
1.3 Research Objective	4
CHAPTER II BACKGROUND	5
2.1 The AES Algorithm	5
2.1.1 SubBytes	6
2.1.2 ShiftRows	7
2.1.3 MixColumns	7
2.1.4 AddRoundKey	8
2.1.5 Side-Channel Leakage	8
2.2 Power Analysis	9
2.2.1 Simple Power Analysis.....	12
2.2.2 Differential Power Analysis.....	13
2.3 Electromagnetic Analysis	17
2.4 Countermeasures.....	19
2.4.1 Evaluating Countermeasures	20
2.4.2 Countermeasure Techniques.....	22

2.5 Circuit Variants	25
2.5.1 Circuit Variant Countermeasures.....	26
2.6 Moving Target Defense	28
2.6.1 Dynamic Partial Reconfiguration	31
CHAPTER III RESEARCH OBJECTIVE	35
CHAPTER IV S-BOX CIRCUIT VARIANTS.....	39
4.1 Program Encryption Toolkit	39
4.2 Side-Channel Properties.....	41
CHAPTER V COUNTERMEASURE DESIGN.....	46
5.1 Equipment and Resources.....	46
5.2 Countermeasure Design	47
5.3 Control Implementation	51
CHAPTER VI SIDE-CHANNEL RESISTANCE.....	52
6.1 Localized EM Analysis.....	52
6.1.1 Results.....	53
6.2 Power Analysis	63
6.2.1 Results.....	64
6.3 Performance and Cost	66
CHAPTER VII CONCLUSIONS AND FUTURE WORK	69
REFERENCES	73
APPENDICES	94
Appendix A: First Order Analysis of Control Implementation	94
Appendix B: First Order Analysis of Countermeasure Implementation	96
BIOGRAPHICAL SKETCH	98

LIST OF TABLES

Table	Page
1. S-box Substitution Values for The Hexadecimal Byte xy [27].	7
2. Research Objectives for Circuit Variant Moving Target Defense.....	37
3. Number of Gates for S-Box Circuit Variants	41
4. Execution Times for AES S-box Variants.....	43
5. LUT Equations for S-boxes 1 and 5.	44
6. Recovered Key from Control Implementation.	62
7. Recovered Key from Countermeasure Implementation.	62
8. Utilization of Control Implementation.....	67
9. Utilization of Countermeasure Implementation.....	67

LIST OF FIGURES

Figure	Page
1. AES Encryption Block Diagram [29].	6
2. Pseudo Code for AES Algorithm [33].	9
3. Block Diagram of a Typical Measurement Setup for Power Analysis [30].	11
4. Power Trace from AES-128 on a Smart Card by Kocher et al. [35].	11
5. SPA Leaks in RSA Modular Exponentiation by Kocher et al. [35].	13
6. Five Steps of DPA Attack based on Mangard et al. [30].	15
7. EM XY Scan at External (Left) and Internal Clock Frequency (Right) [45].	19
8. Iterative Selection/Replacement	40
9. Merged Signature Circuit Generation. 5 Gates with a Max Fan In of 2.	41
10. DONT_TOUCH Logic For S-box 1.	43
11. EM Trace of S-box 0.....	43
12. EM Trace of S-box 5.....	44
13. IP Core Block Diagram.....	48
14. System Block Diagram.	49
15. Countermeasure Device Layout.....	50
16. Control Device Layout.....	51
17. Electromagnetic Analysis Setup.	54

18. Control Hotspot A and Corresponding EM Trace.	54
19. Control Hotspot B and Corresponding EM Trace.	55
20. Control Hotspot C and Corresponding EM Trace.	55
21. Countermeasure Hotspot A and Corresponding EM Trace.	56
22. Countermeasure Hotspot B and Corresponding EM Trace.	56
23. Countermeasure Hotspot C and Corresponding EM Trace.	56
24. Control Correlation for Input Bits 0-3.	57
25. Countermeasure Correlation for Input Bits 0-3.	58
26. Control Autocorrelation Graph.	59
27. Countermeasure Autocorrelation Graph.	59
28. Control First Order Analysis Examples.	61
29. Countermeasure First Order Analysis Examples.	62
30. Power Analysis Setup.	63
31. Control Power Trace.	64
32. Filtered Control Power Trace.	64
33. Control Power Autocorrelation Graph.	65
34. Filtered Control Power Autocorrelation Graph.	66

LIST OF ABBREVIATIONS

AES	Advanced Encryption Standard
PA	Power Analysis
SCA	Side-Channel Analysis
EM	Electromagnetic
RDI	Random Delay Insertion
EMA	Electromagnetic Analysis
SNR	Signal-to-Noise Ratio
ECC	Elliptic Curve Cryptography
CV	Circuit-Variant
CPA	Correlation Power Analysis
P&R	Placement & Route
PL	Programmable Logic
ISR	Iterative Selection/Replacement

ABSTRACT

Mullins, Tristen, H., Ph.D., University of South Alabama, May 2022. Circuit-Variant Moving Target Defense for Side-Channel Attacks on Reconfigurable Hardware. Chair of Committee: Todd R. Andel, Ph.D.

With the emergence of side-channel analysis (SCA) attacks, bits of a secret key may be derived by correlating key values with physical properties of cryptographic process execution. Power and Electromagnetic (EM) analysis attacks are based on the principle that current flow within a cryptographic device is key-dependent and therefore, the resulting power consumption and EM emanations during encryption and/or decryption can be correlated to secret key values. These side-channel attacks require several measurements of the target process in order to amplify the signal of interest, filter out noise, and derive the secret key through statistical analysis methods. Differential power and EM analysis attacks rely on correlating actual side-channel measurements to hypothetical models.

This research proposes increasing resistance to differential power and EM analysis attacks through structural and spatial randomization of an implementation. By introducing randomly located circuit variants of encryption components, the proposed moving target defense aims to disrupt side-channel collection and correlation needed to successfully implement an attack.

CHAPTER I

INTRODUCTION

1.1 Motivation

Several factors are considered when deciding on which platform to implement a cryptographic algorithm. There are many trade-offs between software and hardware implementations including cost, speed, and flexibility. While software implementations are often more flexible, easier to update, and have low development costs, they can have greater overhead costs and weaker security than their hardware counterparts.

Reconfigurable hardware devices (e.g., field-programmable gate arrays, or FPGAs) feature characteristics that allow them comparable flexibility to software implementations while incorporating the benefits of hardware realization. Wollinger and Paar [1] list some potential advantages reconfigurable hardware provides for cryptography including algorithm agility, algorithm upload, architecture efficiency, resource efficiency, algorithm modification, throughput, and cost efficiency. Not only do these improve algorithm performance, but they also ensure that the platform resources are used efficiently, and updates are easily made through reconfiguration. However, these advantages can only be exploited if security shortcomings are addressed.

The security of cryptosystems involves preventing an attacker's ability to obtain information about plaintext. Traditionally, this has been done by prioritizing secrecy of

the key through complex key selection and secure key exchange [2]. With the emergence of side-channel analysis (SCA) attacks, bits of a secret key may be derived by correlating key values with physical properties of cryptographic process execution. Information such as timing [3], power [4], and electromagnetic (EM) radiation [5] side-channel properties can all be observed during run-time of a cryptoprocess. These signals reflect data-dependent system behaviors that may be analyzed by an attacker to derive secret key values.

The ability to obtain information about the system is dependent on the accessibility of a “usable” side-channel and does not “reflect inherent weaknesses” of the process being examined [6]. Therefore, countermeasures for SCA attacks should focus on reducing trace usability by minimizing behavior-key correlation and information leakage within the signal.

1.2 Problem Statement

Side-channel countermeasures are designed to increase the complexity of SCA attacks. This is often done through hiding and masking techniques such as random delay insertion (RDI) [7]–[9], shuffling [10]–[13], masking [14], [15], dual-rail logic [16]–[19], etc. While increasing attack complexity makes it more difficult for an attacker to successfully obtain the key, it does not make it impossible. Because there is currently no solution that eliminates all side-channel leakage, we must “accept that cryptographic implementations leak a certain amount of information,” and avoid allowing the leakages to completely compromise security during use [20].

Many researchers suggest implementing countermeasures in combination to further improve security [12], [21]. Such integration has been used to address the shortcomings of individual countermeasures. For example, masking schemes are often applied in combination with shuffling countermeasures to increase the number of required attack traces [12], [13].

Attackers may also have the ability to perform multiple types of SCA on devices. With physical access to a device, both power and electromagnetic analysis (EMA) attacks may be conducted with simple equipment. Though many power countermeasures are assumed to protect against EMA, it has been shown that power countermeasures may still be vulnerable to localized EMA attacks [22]–[24]. This creates a need for both power and EMA attack prevention methods on a device. To provide sufficient security, designs should include countermeasure combinations that not only protect against single side-channel attacks but alternatives that may be available to an attacker with physical device access.

However, selecting which countermeasures to apply should not be done arbitrarily. Not only is it costly to implement multiple security measures, but some combinations may also add deficiencies to a system. It has been shown that methods used to secure an encryption algorithm against one kind of attack may consequently leave it vulnerable to another [25], [26]. Therefore, it is important to assess the compatibility of countermeasures so that attack vectors are not introduced or aided by their integration.

1.3 Research Objective

This research focuses on increasing the complexity of localized EM SCA by introducing structural and spatial randomization of the target hardware. This is done by utilizing randomly placed S-box circuit variants in the programmable logic side of an SOC. A practical countermeasure evaluation is performed by collecting power and localized EM traces, determining which trace sets are usable, and performing first order differential analysis on those traces.

The remainder of this document is organized as follows. Chapter II provides a background on power and EM side-channel analysis as well as countermeasures. This includes literary review of works involving moving target defense, circuit variants, and dynamic partial reconfiguration. Chapter III describes the goals and objectives of this research while Chapter IV details a proposed methodology for developing and assessing the circuit-variant moving target defense countermeasure.

CHAPTER II

BACKGROUND

In this chapter, a background on power and electromagnetic side-channel analysis is provided. This includes an overview of techniques for attacks as well as resistance. A description of the algorithm used in this research, AES, is also provided. Further, methods for quantifying countermeasure effectiveness are also discussed as well as the applicability of power countermeasures as an EM analysis defense. Related works involving circuit variance, moving target defenses, and dynamic partial reconfiguration are discussed.

2.1 The AES Algorithm

The Advanced Encryption Standard (AES) is the current standard for encrypting electronic data [27]. This symmetric block cipher is a form of the Rijndael cipher [28] that processes 128-bit blocks with variable key length. Each data block consists of 16 bytes arranged in four rows and four columns. The cipher supports key lengths of 128, 192, and 256 bits which correspond to 10, 12, and 14 rounds, respectively.

Figure 1 shows a block diagram for AES encryption. After the initial round key addition, a round function is implemented either 10, 12, or 14 times depending on the key length. Each round consists of four transformations: SubBytes, ShiftRows, MixColumns,

and AddRoundKey. The only exception is the final round which does not perform the MixColumns operation.

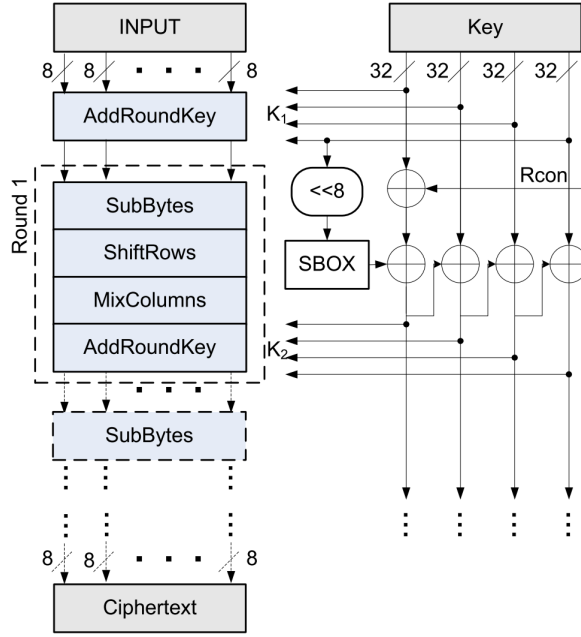


Figure 1. AES Encryption Block Diagram [29].

2.1.1 SubBytes

The SubBytes transformation is a non-linear byte substitution that operates on each byte independently using a substitution table, or S-box. The S-box is constructed by taking the multiplicative inverse in the finite field $GF(2^8)$ and then applying an affine transformation over $GF(2)$ [27]. An example of a S-box in hexadecimal form is shown in Table 1. An input byte xy results in an output byte that is determined by the value at the intersection of row x and column y . For example, an input byte $xy = \{b1\}$ would yield the output byte $\{c8\}$.

Table 1. S-box Substitution Values for The Hexadecimal Byte xy [27].

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

2.1.2 ShiftRows

In the Shift Rows transformation, bytes in the last three rows of the block are cyclically shifted [27]. The bytes in each row are rotated to the left a certain number of times depending on which row they are in. The first row does not shift, the second shift by one, the third by two, and the fourth by three.

2.1.3 MixColumns

The MixColumns transformation operates on each column individually. Each column is treated as a four-term polynomial over $GF(2^8)$ and is multiplied with a fixed polynomial [27]. The MixColumns transformation is not performed in the final round of AES.

2.1.4 AddRoundKey

For the AddRoundKey transformation, a simple bitwise XOR operation is used to add a round key to the data [27]. The round keys are derived from the cipher key using a key schedule that consists of the key expansion and round key selection [28]. The number of round key bits is equal to the block length, N_b , multiplied by the number of rounds plus one. For example, a 128-bit block length and 10-round implementation would require a round key of 1048 bits. The cipher key is used to generate an expanded key. The first N_b words of the expanded key are used for the first round key, the second N_b words for the second round key, etc.

The pseudo code for the AES algorithm is shown in Figure 2. N_r represents the number of rounds. For 128-bit AES, $N_r = 10$. The data block size in words is represented by N_b . Array $w[]$ contains the key schedule. As shown in Figure 2, all N_r rounds of the cipher are identical with the exception of the final round. The final round is executed outside of the for loop and does not include the MixColumns transformation.

2.1.5 Side-Channel Leakage

Of the four transformations in the round function, the SubBytes and AddRoundKey operations are the most prone to side-channel leakage. Any operations with output directly related to the secret key are of particular interest to attackers. SubBytes applies a function to each byte of the state. Therefore, each output byte of the SubBytes transformation in the first round can be calculated based on one byte of plaintext and one byte of the key [30]. While simple countermeasures may be used to mask the side-channel leakage of the AddRoundKey function, the non-linearity of the SubBytes transformation makes it difficult to mask [31]. Further, masked S-box

implementations may still leak information via glitches when realized in hardware, requiring the inclusion of additional countermeasures [32].

```

Cypher
INPUT: byte  $in[4 * N_b]$ , word  $w[N_b * (N_r + 1)]$ 
OUTPUT: byte  $out[4 * N_b]$ 

byte  $state[4, N_b]$ 
 $state = in$ ;
AddRoundKey( $state, w[0, N_b - 1]$ )

for  $round = 1$ , step 1 to  $N_r - 1$  do
    SubBytes( $state$ )
    ShiftRows( $state$ )
    MixColumns( $state$ )
    AddRoundKey( $state, w[round * N_b, (round + 1) * N_b - 1]$ )
end for

SubBytes( $state$ )
ShiftRows( $state$ )
AddRoundKey( $state, w[N_r * N_b, (N_r + 1) * N_b - 1]$ )

 $out = state$ 

```

Figure 2. Pseudo Code for AES Algorithm [33].

2.2 Power Analysis

Kocher et al. first introduced power analysis attacks in 1998 [34]. Their work demonstrated that secret keys may be revealed through power consumption measurements of devices. This method of secret key derivation is based on the behavior of semiconductor logic. When charge is applied to or removed from transistors, a current is induced that consumes power and emits EM radiation [34]. This switching activity may vary depending on which operations are being performed on a device as well as the data

being processed. For example, operations that are performed using different circuits would have differing power consumption behavior. Similarly, varying numbers of transistors may experience switching activity depending on the input values being used [35]. These trends in power consumption may be measured by an attacker and used to determine runtime information that may otherwise be assumed to be private. If the device under observation is executing cryptographic processes, the data-dependent power usage may expose the secret key.

For power analysis attacks, one or more traces must be collected. A trace consists of measurements taken during the execution of the crypto-process being targeted. Mangard et al. [30] presents a block diagram of a typical measurement setup that includes the sequence of interactions for acquiring a power or EM trace. Figure 3 shows this block diagram. In the first step, the cryptographic device is supplied with power and a clock signal. The power measurement circuit or EM probe is also placed during this step while the oscilloscope is initialized in the second. During step 3, commands are sent to the device to start execution. Power consumption is measured at step 4 using the measurement device (i.e., circuit or probe) and oscilloscope set up previously. Power measurement circuits often consist of a small resistor in series with the power supply or ground of the device. The oscilloscope samples the voltage drop across the resistor which is proportional to the power consumption under a constant power supply. EM probes serve as contactless alternatives to power measurement circuits. Using these devices, an oscilloscope measures the output voltage of the probe which is proportional to the device's EM field-inducing power consumption. The PC receives the output of the

cryptographic process in step 5 and the power trace from the oscilloscope in step 6. To collect multiple traces, steps 2 through 6 are repeated as necessary.

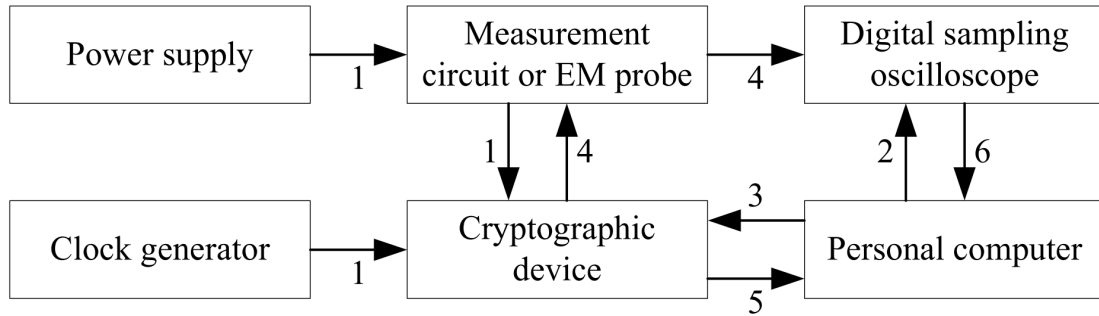


Figure 3. Block Diagram of a Typical Measurement Setup for Power Analysis [30].

Figure 4 shows an example power trace for AES-128 encryption on a smart card collected by Kocher et al. [35]. The 10 rounds of AES are clearly visible within the trace, a characteristic that can aid an attacker in identifying which encryption algorithm their target is using.

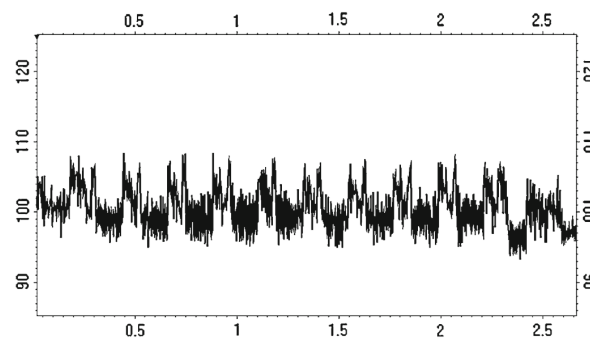


Figure 4. Power Trace from AES-128 on a Smart Card by Kocher et al. [35].

Simple implementations that yield low-noise measurements may be broken with a single trace in a Simple Power Analysis (SPA) attack. The more complex the device, however, the noisier the signal may be. Electronic noise from the power supply and clock generator or switching noise from other components and parallel operations may decrease the signal-to-noise ratio (SNR). A low SNR can be overcome by taking more traces as well as utilizing signal processing methods. Differential Power Analysis (DPA) may be applied in scenarios where SPA falls short. DPA attacks utilize statistical functions that are designed for specific cryptographic algorithms [36].

2.2.1 Simple Power Analysis

With simple power analysis attacks [34], the attacker attempts to infer information about a process directly by visually analyzing a single power trace or very few traces. I/O operations and individual rounds have been identified using SPA profiling [35], [37]. Biham and Shamir introduced a profiling method to identify key scheduling [37]. Using their method, data-dependent portions of the trace are found by comparing power traces that process different input data. Key-dependent portions are identified using the traces from multiple devices which each have a unique key. Several models link power consumption to the hamming weight of the processed data or the hamming distance between that data and [38]–[41]. This information leakage that is observable using SPA, can significantly reduce the number of candidates when trying to brute-force search the key [42]. When no countermeasures are implemented, data-dependent instruction sequences may reveal power consumption differences for “0” and “1” key bits. Kocher et al. demonstrates this using a simple implementation of RSA, shown in

Figure 5 [35]. This trace was gathered during the modular exponentiation step of RSA in which squares and multiplies are performed using bits from the decryption key. Per the structure of the algorithm, squares are consumed in each iteration of the loop while multiplications are only performed when the bit of the exponent is equal to 1. This behavior can be directly observed in the power consumption of the unprotected device since the multiplication operation consumes more power than the square, allowing the bits of the decryption key to be identified. In order to find the key, the attacker must have detailed knowledge of the algorithm used by the target. Nevertheless, by revealing data and operation dependent power consumption with minimal traces, SPA techniques may still be leveraged by attackers to aid in more complex SCA attacks against protected implementations.

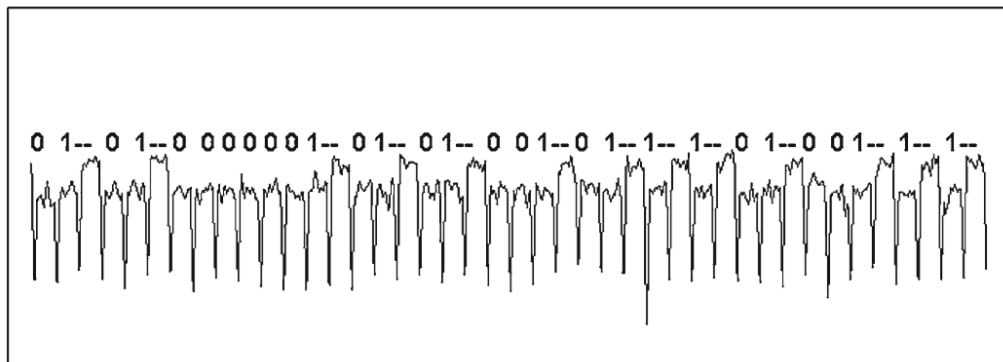


Figure 5. SPA Leaks in RSA Modular Exponentiation by Kocher et al. [35].

2.2.2 Differential Power Analysis

Differential power analysis [34] uses algorithm-specific statistical methods to identify data-dependent correlations in power traces. These attacks differ from SPA

attacks in several ways. Where SPA analyses a single trace over time, DPA requires a large number of traces and is able to find small correlations at specific points [30]. The attacker often does not need to be knowledgeable of details beyond which algorithm is used by the target to perform a DPA attack as opposed to SPA.

DPA attacks all follow a general procedure. Mangard et al. [30] describe the DPA strategy in 5 steps which are displayed in Figure 6.

2.2.2.1 Step 1: Select Intermediate Value. In the first step, the attacker must choose an intermediate result on which to base the attack. This result must be a key-dependent value (e.g., an XOR operation or S-box output for AES) and may be represented as a function of d and k where d corresponds to plaintext or ciphertext and k is the key.

2.2.2.2 Step 2: Collect Traces. Power consumption measurements are taken in step 2. The attacker must know each data value d that is processed, represented as vector $\mathbf{d} = (d_1, \dots, d_D)'$. For each of the D data blocks, a power trace t is taken at each encryption or decryption, i . The power trace for d_i is $\mathbf{t}'_i = (t_{i,1}, \dots, t_{i,T})$, where T is the length of the trace. The resulting power traces for step 2 are shown in Figure 6 as a matrix of size $D \times T$ and all use the same secret key.

2.2.2.3 Step 3: Calculate Hypothetical Intermediate Values. The vector $\mathbf{k} = (k_1, \dots, k_K)$ is comprised of all K key hypotheses. Each of these elements are used in the calculation of hypothetical intermediate values $f(d, k)$ in step 3. The results are found in matrix \mathbf{V} of size $D \times K$ where $v_{i,j}$ represents the intermediate value corresponding to d_i and k_j . Because \mathbf{k} includes all possible values for the key, one column in \mathbf{V} corresponds to the intermediate value that was calculated using the correct key.

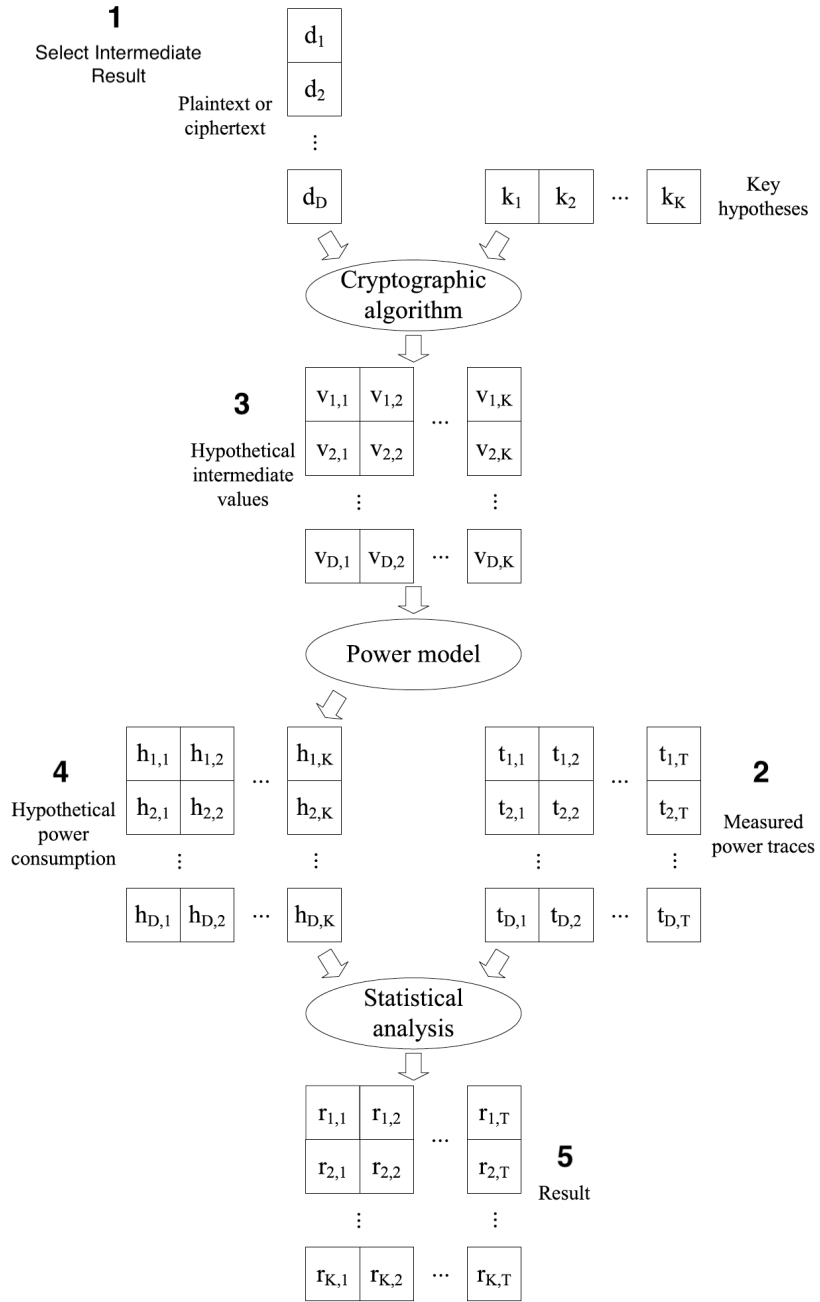


Figure 6. Five Steps of DPA Attack based on Mangard et al. [30].

2.2.2.4 Step 4: Map Hypothetical Power Consumption. Hypothetical power consumption, matrix \mathbf{H} in Figure 6, is then obtained for each intermediate value v in step

4. This is done using simulation techniques that are based on the attacker's understanding of the target device's behavior. Hamming-distance and Hamming-weight models are the most common power models used in DPA attacks due to their ease of application [30]. Customized power models increase the effectiveness of the attack but are up to the attacker to derive using their knowledge of the device.

Before performing the final step of a DPA attack, the attacker needs to make sure each column \mathbf{t}_j consists of similar operations before calculating the correlation coefficients for matrix \mathbf{R} . This can be done using a trigger signal to indicate the beginning of a specific operation and initiate measurement with the oscilloscope. In a controlled setting, the attacker would be able to program the device to trigger the oscilloscope consistently before a process. However, an attacker may not have sufficient control over the device for this method. If no other useful signals can be measured from the device, alternatives may be used such as the start signal from the PC to cryptographic device shown in step 3 of Figure 3. Using such asynchronous signals may result in inconsistent delays between the trigger and start of encryption. Attackers must utilize alignment methods when preprocessing their traces to remove these delays and ensure that the power consumption within each column \mathbf{t}_j is dependent on the same operations.

2.2.2.5 Step 5: Comparison of Power Consumption. Finally, in step 5, the hypothetical and actual power consumption for each key hypothesis is compared. That is, the columns of matrix \mathbf{H} are compared to the column of matrix \mathbf{T} to obtain a matrix of size $K \times T$, \mathbf{R} . Each element $r_{i,j}$ corresponds to the correlation coefficient of columns \mathbf{h}_i and \mathbf{t}_j and range in value from -1 to 1. An explanation of the correlation coefficient algorithm can be found in [30]. The attacker assumes that there exists a column \mathbf{h}_{ck} that

corresponds to the hypothetical power consumption calculated using the correct key. It is also assumed that there exists a column \mathbf{t}_{ct} that contains power consumption values that depend on the intermediate values selected in step 1. These two columns yield the highest value in matrix \mathbf{R} , $r_{ck,ct}$. The location of this element in \mathbf{R} reveals the correct key hypothesis as well the position of the power trace at which intermediate values are processed. If there is no clear maximum value $r_{ck,ct}$, more traces may need to be taken to determine the relationship between the columns of \mathbf{H} and \mathbf{T} .

2.3 Electromagnetic Analysis

While data-dependent current flow serves as the basis for power analysis attacks, it also emits electromagnetic fields that can contain key-revealing information. Simple and Differential Electromagnetic Analysis attacks (SEMA and DEMA, respectively) follow similar statistical analysis methods to SPA and DPA using signals collected from EM field probes [35], [36], [43]. Using EM rather than current-based power measurements for attacks does have its advantages. EM measurements offer a desirable alternative to power consumption when access to the power and ground lines are limited, when the power signal contains too much noise, or when power analysis countermeasures are implemented [43]–[46].

The majority of difference between EMA and power analysis methods are a result of the respective signals' frequency contents. Debeer et al. [45] identify four points of variation: aliasing, alignment, resampling, and probe positioning. Unlike power signals, EM signals maintain their strength at high frequencies. Because of this, samples taken at too low a frequency could misrepresent the original signal, a concept referred to as

aliasing. This may be prevented by excluding high frequencies during trace collection using a low-pass filter between the probe and oscilloscope. Debeer et al. [45] also describe how common techniques for alignment and resampling of power traces are not applicable to unprocessed EM traces due to their lack of low-frequency components, but can be used after some simple preprocessing methods.

When using a high-resolution EM probe, attackers may perform localized attacks using traces collected from a specific area on the chip [44]. These types of attacks are most successful when the probe is placed above the area of the chip where the side-channel leakage of interest is strongest [47]. To determine the optimal probe position, measurements are taken across the surface of the chip during the execution of the target process. If the chip hosts a variety of processes with distinct clock frequencies, the EM signal should be filtered to reduce components that are not related to the target operations. Figure 7 shows how the location of the strongest signal is dependent on the clock frequency observed. This XY scan of a smart card by Debeer et al. [45] displays EM signal strengths that indicate the location of the main processor and the crypto-processor which run at the external and internal clock frequency, respectively. The location with the strongest signal at the target process's clock frequency indicates the optimal probe position for trace collection. If there are multiple locations that meet this criteria, the position with EM signal behavior that can be related to the target process should be selected [45]. Other works have also identified leakage "hotspots" by performing EM attacks at multiple locations on a chip and plotting the correlation values in a heat map [48], [49].

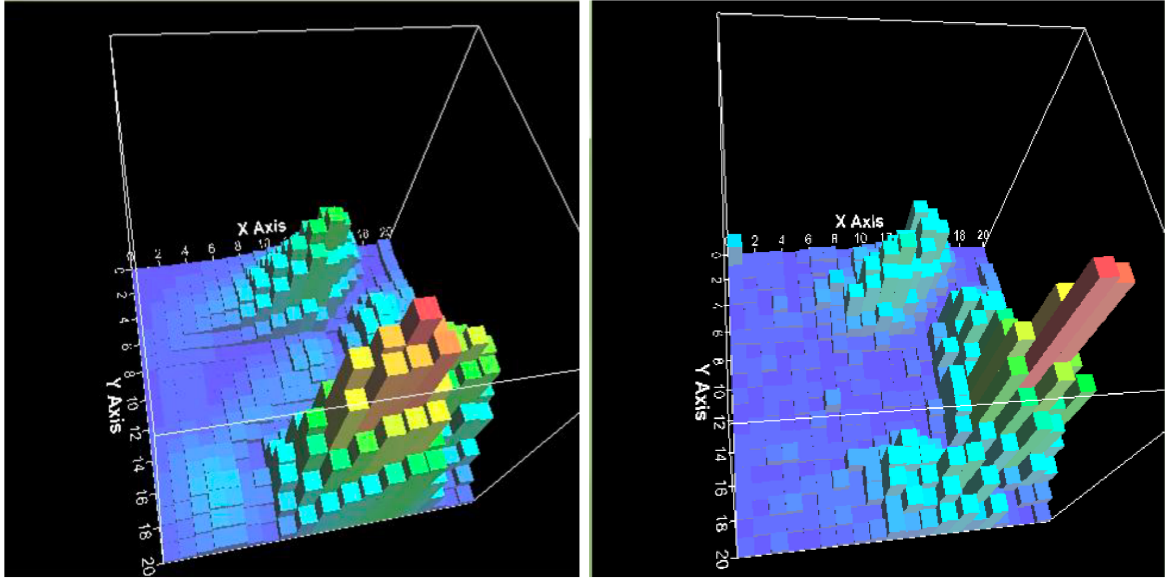


Figure 7. EM XY Scan at External (Left) and Internal Clock Frequency (Right) [45].

After traces are collected and processed, the statistical analysis for EMA attacks is similar to those of SPA and DPA. In their 2001 work, Quisquater and Samyde [43] attribute this to EM signals containing “at least the same information” as power signals. Agrawal et al. [22] and Gandolfi et al. [44] found that EM emanations contain multiple information leakages that can be used in attacks where SPA/DPA may fall short. Other works comparing leakage models have also found EM signals to contain more information than current-based power signals [50], [51].

2.4 Countermeasures

Side-channel countermeasures aim to minimize information leakage as much as possible. Because leakage cannot be entirely eliminated [20], designers need a method of quantifying how effective their countermeasures are against attacks.

2.4.1 Evaluating Countermeasures

Evaluation is an important part of conveying the impacts of countermeasure designs. This is often done using one of two approaches: proving a countermeasure in theory or in practice [52].

Many countermeasures have been shown to be theoretically secure through mathematical proofs [13], [53]–[57]. While leakage may be sufficiently minimized under the assumptions of the proof, the design may still be vulnerable to attacks when realized on a physical device. The models used for these theoretical security evaluations require assumptions that are not able to consider all possible leakage sources. For example, several mathematically secure masking schemes have been found to leak information via logic gate switching activity or hardware glitches [58]–[60]. In their 2012 work [52], Moradi and Mischke further evaluate Prouff and Roche’s glitch-free masking scheme [57]. They found that though the scheme was secure under the assumptions of the original article, more realistic analysis revealed exploitable leakages that were out of the scope of the original model. The authors suggest that proposals supported by theoretical security proofs may leverage the real-world perspective of practical analysis to obtain a more thorough security evaluation. Using practical evaluation methods to support the claims of theoretical security proofs has since become more common in the literature [61]–[64].

Practical countermeasure evaluation is done using real side-channel measurement traces. Many researchers quantify the effectiveness of proposed countermeasures using the number of traces needed to break the encryption. This metric provides insight on how much resistance a countermeasure provides against a specific attack. Number of attack

traces may be used to evaluate how a countermeasure compares to an unprotected implementation in a controlled setting. This method may also be used to evaluate incremental changes to schemes such as introducing additional countermeasures for a combined security approach [12], [65], [66]. There are limitations to comparing proposed countermeasures from distinct works using this metric since number of required traces is dependent on several variables including target device, equipment used for trace collection, and analysis methods. Because differential SCA attacks rely on a device-dependent power model, the number of required traces is not necessarily guaranteed for all implementations. It is important that this metric only be used to compare countermeasures that have been implemented on similar devices. Furthermore, a countermeasure that guarantees security within a given number of traces against one attack does not necessarily guarantee the same level of security against other side-channel attack methods [12], [67]. Therefore, it is important for a researcher to make clear the scenarios in which the countermeasure may achieve the presented level of security.

The cost of implementing a countermeasure is another aspect that should be considered by designers. When a user is selecting among effective and usable countermeasures, the security-cost trade-off may be the deciding factor. Many researchers describe countermeasure costs in terms of number of specific logic elements. However, hardware requirements for a given implementation may vary among devices. In [68], Katashita et al. show that the lookup table specifications for two FPGAs result in large resource utilization differences for the same circuit. Similar to the security metrics discussed earlier, resource utilization may be used to describe costs relative to an unprotected implementation but have limitations due to their device-dependency. Other

cost metrics used to evaluate countermeasures include performance and storage overhead [49], [66], [69].

2.4.2 Countermeasure Techniques

Side-channel analysis countermeasures focus on minimizing the correlation between key-dependent operations and the data that is leaked. Techniques traditionally fall into one of two categories: hiding and masking. In this section, both categories are discussed as well as methods specific to EMA prevention.

Hiding countermeasures involve decreasing the signal-to-noise ratio for a side-channel. This is often done through leveling techniques which decrease the signal or randomization techniques that increase noise level [70]. Many leveling techniques have been developed to minimize key-dependent fluctuations in power consumption and execution timing. Some examples include using low-drop-out voltage regulators to maintain a constant voltage across encryption blocks [71], dual-rail pre-charge logic to control the number of observed transitions on a power side-channel [72], and constant-time operation to thwart timing attacks [73], [74]. Some randomization techniques include shuffling operations [75], insertion of dummy rounds [76], and random delay insertion [77] to introduce noise to hinder power and timing attacks. Some reconfigurable hardware has the advantage of being able to implement countermeasures that leverage dynamic reconfiguration capabilities. This feature of the device can be used to introduce randomness in timing, target location [66], and hardware structure [78]. Though all of these countermeasures increase resistance to side-channel attacks, they would need to be implemented in combination with other countermeasures to sufficiently reduce the risk of leaking key-revealing information [12], [65], [76], [79].

One of the most common countermeasures for side-channel attacks is masking. This method involves randomizing intermediate values so that there are no dependencies between side-channel info and the actual secret key. This countermeasure is implemented at the algorithm level and includes methods such as Boolean [56], [78], [80], multiplicative [81], [82], and combinations of the variants [83], [84]. Like hiding schemes, masking countermeasures are also often implemented in combination with other countermeasures to reduce side-channel related risks [10], [12], [85]. When masking schemes are realized in hardware, logic gate switching activity (i.e., glitches) can leak information that could be leveraged by a side-channel attack [58]–[60]. To overcome this risk, masking schemes for hardware should be designed to either work in the presence of glitches [86] or avoid them altogether [18], [87].

Many researchers consider EM analysis a variant of power analysis, grouping methods and countermeasures for both [35], [88]. The similarities between the side-channel types allow for their countermeasures to be inclusive against standard attacks. However, it has been shown that implementations with DPA countermeasures may be vulnerable to more advanced localized EM attacks [23], [24]. In [23], Specht et al. use localized EMA to isolate the leakage from separate shares in a threshold implementation countermeasure. Their attack combines leakage from multiple probes to break the scheme. Another example is dual-rail logic which has been shown to prevent power attacks becomes vulnerable to localized EM attacks due to placement and routing imbalances [24]. The shortcoming of power countermeasures against localized EMA attacks highlights the need for an additional family of countermeasures.

Existing countermeasures against EMA fall into two categories similar to masking and hiding: signal information reduction and signal strength reduction [22]. Signal information reduction involves randomization and refreshing techniques that are also used for power SCA defense (e.g., additional noise, masking). Signal strength reduction includes techniques that are unique to EMA prevention such as spatial randomization and shielding.

In [66], Mentens et al. introduce “spatial jitter” which randomizes the location of functional blocks dynamically to prevent EMA. Li et al. [48] propose a spatial randomization of dataflow in which data bytes are randomly assigned to AES S-boxes that are places throughout the FPGA fabric. By randomizing the location at which the target logic block is placed, EM leakage hotspots are reduced and the optimal probe location for the attack is difficult to determine.

While traditional SCA countermeasures aim to reduce the usability of a captured signal, shielding techniques work to prevent signal capture altogether. In [89], Das et al. propose a Signature Attenuation Embedded CRYPTO with Low-Level metal Routing (STELLAR). Their technique prevents leakage through EM radiation by routing the design to low-layer metals as well as including signature attenuation hardware to hide the signal. Shielding designs have been proposed that also include an anti-tampering mechanism [90], [91]. The shield utilizes substrate layers that allow for the conduction of an integrity related signal that is broken when the shield is removed. This method was initially proposed as a method to prevent fault injection attacks but could also be used to prevent an attacker from removing shields intended to block EM radiation if incorporated by vendors as suggested in [92]. Miura et al. [93] propose an EM attack sensor which

detects when a near-field EM probe approaches the chip. This concept is based on previous work which demonstrates that a probe cannot measure the original EM field without disturbing it [94]. Though shielding concepts are effective at limiting EM attacks, they incur high packaging costs. The inclusion of detection mechanisms is also accompanied by additional overhead.

2.5 Circuit Variants

Circuit variants refer to designs that are structurally different but have similar functionality. These designs may be diverse in logic gate types, size, and include redundancies all of which result in variations in path delays and consequentially, side-channel behavior.

The goal of delay-based countermeasures is to reduce the ability to align traces collected by an attacker [95]. Randomizing the timing behavior of an implementation results in a desynchronization effect that introduces noise within the trace set [7]. The ability to align the portion of the traces being targeted is a crucial step in successfully performing a differential analysis attack. If a trace is unable to be aligned with a selected reference trace within the set, it is discarded. If a large portion of the traces within the set are discarded, the attack cannot be reliably performed. Delay-based countermeasures have been implemented in both software and hardware schemes through random delay insertion, random process interrupts, and temporal jitter [7]–[9], [21], [96].

Delay characterizations may also be observed and exploited at the circuit level of designs. In combinational logic, propagation delay refers to the maximum time it takes for an output to reach its final value after an input switch and is the sum of the delays

through each element on the critical path [97]. The delay characteristics of logic gates can vary from nanoseconds to the picosecond range depending on the technology being used [98]. Therefore, the timing behavior of a circuit may be directly influenced by the types of gates used to construct it as well as the number of gates. Existing works have shown that variations in gate compositions are translated to the timing behavior of a circuit which may be leveraged in side-channel countermeasures [88].

2.5.1 Circuit Variant Countermeasures

In 2003, Benini et al. introduce the concept of mutating a data path using power-masked modules [99]. This scheme combines a fully functional unit A and a smaller block B that implements the most typical behavior of A but consumes less power. By activating block B rather than block A when inputs allow, the same functionality can be obtained with a randomized power profile. This concept was later extended to reconfigurable hardware by Stöttinger et al. to protect AES [100]. Their approach shuffles modules to tamper with correlation between real and estimated power consumption levels, thwarting DPA attacks.

In [88], Bow et al. utilize two methods of circuit variance for their countermeasure: synthesis-directed and circuit-directed. The synthesis-directed technique involves generating netlists for S-boxes based on a behavioral description and a standard cell library. For each netlist, the available logic gates for the standard cell libraries were changed, forcing the synthesis tool to utilize different logic gates for the implementation. In the circuit-directed technique, a clock delay circuit is used to add random delays along paths within the design. Three synthesis-driven implementations with S-boxes at different

locations for each are combined with three circuit-driven implementations to obtain twelve static versions of the AES engine. Correlation power analysis (CPA) attacks are applied to a trace set composed of measurements from all twelve AES versions. The scheme improved resistance to CPA by more than two orders of magnitude over unprotected AES. Leakage present in the circuit-directed variants suggest that this method alone would not provide sufficient protection against power analysis attacks. However, the fully synthesis-directed approach is limited to only three versions.

Hettwer et al. use a similar synthesis-directed technique for generating diversity [49]. For each variant, 80% of the slices for the defined reconfigurable area are prohibited for placement until after the other 20% has been placed. This method is used to create 128 versions of the AES engine that are randomly selected for configuration. The placement and route (P&R) restrictions for the bitstream generation enabled a spatially randomized design that is effective against localized EMA and fault injection. However, resistance to CPA is only improved by a factor of 2-3, requiring additional countermeasures to sufficiently prevent such attacks.

The lack of power resistance of the approach in [49] may be attributed to the method of variant generation. By only implementing P&R specifications, it is likely that there is minimal diversity within the bitstreams in terms of gate composition. A lack of diversity in this context would result in similar power profiles between the implementations even though they are mapped throughout the FPGA fabric. Therefore, extraction of the key is still possible when an attack is performed using the power side-channel rather than localized EM. The SPREAD approach [88] was able to achieve a much higher level of resistance with significantly fewer versions of AES. This may be

attributed to the gate-level adaptations provided by the synthesis- and circuit-driven techniques used. It is possible that the countermeasure presented in [49] could be improved using different circuit variant generation schemes (e.g., those used in [88]), but diversity would be limited by the size of the circuit since the entire AES core is replaced.

2.6 Moving Target Defense

Cyber defense includes three complimentary categories: proactive, active, and regenerative [101]. In this cohesive model, proactive measures harden the system to make it more resilient against attacks, active defenses involve attack detection and real-time responses, and regenerative techniques are used to restore the system after an attack. Each of these techniques are reactive in nature, designed to patch known vulnerabilities of a system or respond to an attack that has been detected [102].

A Moving target defense (MTD) is a more proactive approach. System changes are made over time to create a varying attack surface [103]. Rather than hardening specific aspects of a configuration, MTD enables a complex target that makes attacks more difficult to complete. Modifying characteristics of the system pseudo-randomly disadvantages the attacker in the reconnaissance phase [104]. The time attackers have to discover and exploit vulnerabilities is limited in an MTD system. Persistence is also more difficult for cyber-attacks since any privileges gained may be lost when the system is altered [105]. MTD techniques may also be used to introduce additional protection to systems in which other security mechanisms are already implemented [102].

MTDs have been introduced at many different levels to protect a variety of systems and devices. Address hopping and port hopping may be used to protect networks

[106]–[108]. Address space layout randomization, data space randomization, and instruction set randomization are deployed in most current operating systems [109]–[111]. MTD techniques have also been used to provide low-cost side-channel attack prevention in several contexts including cloud architectures [112], processor caches [113], embedded systems [69], [114], among others [115].

Many cryptographic SCA countermeasures have also incorporated MTD methods through refreshing parameters. Masking schemes, for example, need to implement a sufficient level of refreshing in order to remain effective [62], [116], [117].

Cryptographic targets may also disrupt SCA by updating the secret key. This concept, called key refreshing, involves generating new session keys from a nonce and master key to thwart SCA attacks. The principal behind using re-keying methods is that the burden of protection is shifted from the cryptoprocess to the easier to secure re-keying algorithm [118]. The rate of key-refreshing determines the window in which an attack can occur. While some works propose a new key for each block of plaintext [119], frequent updates can introduce significant costs to the system since the nonce would need to be synchronized across all parties [114]. It is also important that the key update function is secure against SCA to prevent the extraction of the master key [120]. Recent works have proposed securely rekeying at an interval that is based on the number of required traces to complete a power analysis or EMA attack [114], [121]. Similar countermeasures involving register renaming [122], [123] and algorithm-level parameter randomization [26], [124]–[127] have been proposed which pseudo-randomly alter characteristics about the system that are leveraged in side-channel attacks.

Dynamic logic reconfiguration (DLR) has been used to implement FPGA-oriented MTDs in which redundant logic blocks are placed throughout the FPGA fabric and are randomly selected for operation at runtime. This method has been used to hinder hardware trojans [128], [129] and provide dynamic side-channel countermeasures in hardware [48], [69].

In their 2019 work, Li et al. [48] propose a DLR-based spatial randomization technique to minimize leakage that may be exploited through localized EM attacks. In this MTD countermeasure, a permutation network is used to randomly assign data bytes to sixteen AES S-boxes and a second permutation network restores the order of the bytes. By using logic gates rather than look-up tables to synthesize S-boxes, the designer is able to select the location for each S-box, allowing maximum distance between each component. Two attack scenarios are simulated: one where the attacker has full access to the device, enabling a profiling attack, and a second black-box attack. Both attacks perform correlation analysis using EM traces. When determining the optimal location of the probe, the profiling attacker observes no distinct hotspots while the black-box attacker observes one. This hotspot is linked to the state registers' location which is unaffected by the spatial randomization in the S-box layer. The countermeasure increases the number of required attack traces by 150X for the profiled attack and 3.25X for the black-box attack. Further work is needed to determine if a similar countermeasure can be applied to reduce the leakage of the state registers. Implementing another countermeasure in combination with the spatial randomization to further increase the number of attack traces is another open area of research.

The dynamic nature of MTDs enables changes in systems that would otherwise be static. However, there exists a family of devices that allow hardware configurations to be altered and/or placed at runtime. This feature, called dynamic partial reconfiguration, supports the implementation of more complex MTDs in hardware.

2.6.1 Dynamic Partial Reconfiguration

Some system-on-a-chip (SoC) and field programmable gate-arrays (FPGA) have the capability to alter portions of their hardware configuration during run-time without interrupting the rest of the chip [130]. This feature, referred to as dynamic partial reconfiguration (DPR) and is different from the previously discussed DLR in that DPR schemes change the placement and routing of functions where DLR functions are static. DPR can be utilized to implement complex moving target defense techniques in hardware designs.

In [66], Mentens et al. propose the first DPR countermeasure to reduce side-channel leakage. Their approach introduces temporal jitter by randomly adding registers between functional blocks to introduce delays. The countermeasure also increases resistance to fault injection by altering the location of functional blocks which is referred to as spatial jitter.

Stöttinger et al. [131] propose a SPA and DPA countermeasure for elliptic curve cryptography (ECC) that combines the techniques presented in [66] and by Benini et al. in [88]. This countermeasure introduced temporal jitter as well as parallel modules that can be dynamically reconfigured with implementation variants. In a later work, Stöttinger et al. [100] adapted the DPA countermeasure presented in [99] for reconfigurable

platforms. After each encryption, the AES countermeasure uses DPR to reconfigure one of the two S-boxes in the FPGA fabric with a new implementation. Each implementation is functionally the same but have different side-channel behavior, reducing the correlation between real and estimated power consumption values.

Similar to the work presented in [48], Bloom et al. [132] propose a scheme in which spatial randomization is used to protect a device. This countermeasure, however, uses DPR rather than DLR to change the location of IP blocks to prevent design- and fabrication-time trojans from attacking fixed structures in the FPGA. Their hardware abstraction layer decrypts bitstreams for new IP cores, finds an unused random location, completes place and route for the core, and deletes the previous one.

Hettwer et al. [49] propose a countermeasure against physical attacks in which the entire AES core is replaced with a circuit variant implementation over time. Though power SCA resistance was only increased by at most a factor of 3, the random changes to the physical layout of the configurations made this countermeasure especially effective against localized EMA and fault injection. Localized EMA was performed at 135 locations across the chip. The highest correlation value obtained using 5000 traces was 0.1 for the reference implementation and 0.06 for the countermeasure. The probability of injecting a fault is also reduced to less than 1% with the countermeasure implemented. The replacement of the entire AES core results in encryption stalls during reconfiguration as well as a large storage overhead for the bitstreams.

Huss et al. [133] describe a concept for a mutating runtime architecture which combines three countermeasures for AES. Recent work [88] presents a similar approach for side-channel power resistance for encryption algorithms using DPR called SPREAD.

This countermeasure utilizes an extra copy of an AES S-box to allow partial bitstream reprogramming. A customized tool is used to create relocatable bitstreams which will allow the same bitstream to be programmed at different locations by changing a frame address. A multiplexing scheme is used to isolate the redundant S-box so that it can be replaced with a functionally equivalent variant of one of the other sixteen S-box structures and reincluded in the AES engine. Another S-box is then randomly selected to be reprogrammed as the process is repeated. The work published in [88] displays a proof of concept in which 12 versions of an AES engine are tested against DPA and CPA. A fully operational version is needed to determine the actual number of traces required to break the implementation. Further research is also needed to determine how susceptible this countermeasure is to localized EMA.

Besides SCA resistance, an important consideration when designing and selecting DPR-based countermeasures is their cost. Even though DPR schemes may have less inactive logic consuming real-estate than DLR schemes, they can still be expensive which can limit their applicability and usefulness [132]. Storage overhead for reconfiguration bitstreams can significantly increase the requirements for an implementation. For example, a partial bitstream for the scheme proposed in [49] has a size of 616 MB, requiring over 700 MB to store all 128 variants. This requires external flash to be able to implement the countermeasure as it is designed. The authors also suggest implementing additional countermeasures to increase power SCA resistance which would further increase the cost. An alternative is presented in [88] in which only S-boxes are reconfigured, but further research is needed to determine how much diversity is sufficient in a fully dynamic scheme. Further, the timing and throughput overhead

should also be considered in the design process. The setup time for DPR schemes as well as any function stalls for reconfiguration can make the security-efficiency trade-off less desirable. When composing a countermeasure involving DPR, minimizing the number of required bitstreams and reconfiguration frequency may lead to a design that is applicable to a wider variety of reconfigurable devices.

CHAPTER III

RESEARCH OBJECTIVE

Related works have shown that individually, spatial randomization and implementation diversity may be used to obfuscate optimal EM probe positions and increase resistance to power analysis attacks, respectively. However, countermeasures that have attempted to combine these concepts have yet to display resistance to both power analysis and localized EMA attacks in an efficient manner. A spatially randomized implementation may hinder an attacker using a high-resolution EM probe, but if the power consumption behavior does not vary between implementations, an attacker may still perform a side-channel attack that is not location-dependent (i.e., a power analysis attack). This scenario is unfavorable since the equipment to perform a power analysis attack is simpler and more affordable than that of a localized EM attack [30], [47]. Therefore, it is in the researcher's best interest to ensure that defenses against localized EM attacks are also resistant to power attacks.

The goal of this research is to increase the complexity of both power and localized EM SCA by introducing structural and spatial randomization of the target hardware. We propose a countermeasure that utilizes randomly located S-box circuit variants in the PL side of an SOC. The focus of this approach is limiting the presence of EM “hotspots” that indicate favorable candidates for high-resolution probe placement as discussed in Section

2.2. One S-box will be selected and used as the output for the encryption run, mimicking the behavior of a MTD, and increasing the number of traces needed to perform a localized EMA attack.

Power analysis resistance will be introduced to the design through the variation in circuit structure and composition of the S-boxes. By diversifying the implementations at the gate-level, we aim to vary the power behavior observed by the attacker and disrupt the correlation between the hypothetical and actual power consumption. For this countermeasure, all circuit variants will be generated using a program encryption toolkit (PET) that allows for multiple criteria to be set including subcircuit selection and replacement size, gate type, and fan-in. The influence the variants have on side-channel behavior will be determined by implementing multiple AES versions, each with a different S-box variant. EM traces will be collected for each and their behavior compared.

A practical countermeasure evaluation will be conducted to determine the implementation's resistance to power and localize EM analysis. Power and EM traces will be collected for both a control and countermeasure implementation. Usable trace sets will be determined by the inclusion of AES artifacts such as repeated round structures visible within the traces. DPA and DEMA attacks will be performed on the usable trace sets using first order analysis. The number of traces needed to obtain sufficient confidence values of key candidates to differentiate between correct and incorrect bytes will be used to quantify the success of the attacks. During the acquisition step of the DEMA attacks, the optimal probe placement will be determined as well as any leakage hotspots identified. These characteristics will be used to investigate the level of hotspot

obfuscation provided by the countermeasure. The objectives of this research are shown in Table 2.

Table 2. Research Objectives for Circuit Variant Moving Target Defense

Objectives	Description
1.	Circuit Variant Side-Channel Behavior Study.
1.1	Generate S-box circuit variants.
1.2	Design control AES implementation.
1.3	Collect EM traces for each AES-S-Box version.
2.	Investigate EM Hotspots.
2.1	Modify AES core to accommodate multiple S-box instances.
2.2	Randomly place S-box P-blocks in programmable logic.
2.3	Collect EM spectralintensity graphs of both implementations.
3.	Assess Trace Sets for Usability.
3.1	Identify AES artifacts within the trace sets.
3.2	Determine the target window for first order analysis.
4.	Differential First Order Analysis Attacks.
4.1	Improve SNR of collected trace sets.
4.2	Evaluate attack success using confidence values for key candidates.

This research differs from related works in both the circuit variant generation specifications and the increased resistance to localized EM attacks. By limiting the design to only altering S-boxes, we expect to reduce the storage overhead of the scheme proposed in [49]. Another distinction from [49] is the criteria for generating variants which will focus on gate-level diversity and circuit size. This is expected to further increase resistance to power attacks. This design aims to implement a DLR scheme where

the SPREAD scheme in [88] utilizes a custom synthesis tool flow to create relocatable bitstreams. Though this may result in less resource usage for an active implementation in SPREAD, the reconfiguration logic may introduce noise in the side-channel. The lack of a reconfiguration controller in the proposed DLR design leaves more area in the PL for S-box variants. This research will also only utilize a synthesis-driven circuit variant generation method as opposed to SPREAD which also includes additional hardware for a circuit-driven approach. To introduce gate-level diversity, we will use a program encryption toolkit [134] to generate equivalent circuits that vary in size and composition where the method in [88] exclude specific gate types when generating netlists for each version. Lastly, the proposed countermeasure will be not only be assessed for power analysis resistance but for localized EM analysis resistance as well.

CHAPTER IV

S-BOX CIRCUIT VARIANTS

This chapter details the generation of AES S-box circuit variants (CVs) using a Program Encryption Toolkit (PET). In particular, the Iterative Selection/Replacement feature of PET is described. The influence the CVs have on side-channel properties for an AES implementation are also studied.

4.1 Program Encryption Toolkit

PET is a customized Java application that includes features to generate random equivalent circuits based on an ICAS BENCH format netlist [134], [135]. The netlist used as the reference circuit for our S-box consisted of a gate-level implementation that follows the behavior of the standard AES SubBytes function shown in Table 1. PET's Iterative Selection/Replacement (ISR) feature was used to diversify subcircuits within the S-box structure. In this method of variant generation, shown in Figure 8, a user specifies a size range for randomly selected subcircuits and a number of iterations for the process [136]. For each iteration, a subcircuit is selected and replaced by a randomly generated equivalent circuit. Characteristics of the random circuit may be set by the user including number of gates, max fan-in, as well as gate types. The circuit variants used in this research were generated using replacements with a max fan-in of two, selection size of

two, replacement size of six, and only excluded BUFFER gates. In addition to the original, five variants were used in the study that were generated using 100 iteration intervals ranging from 100 to 500 ISR iterations.

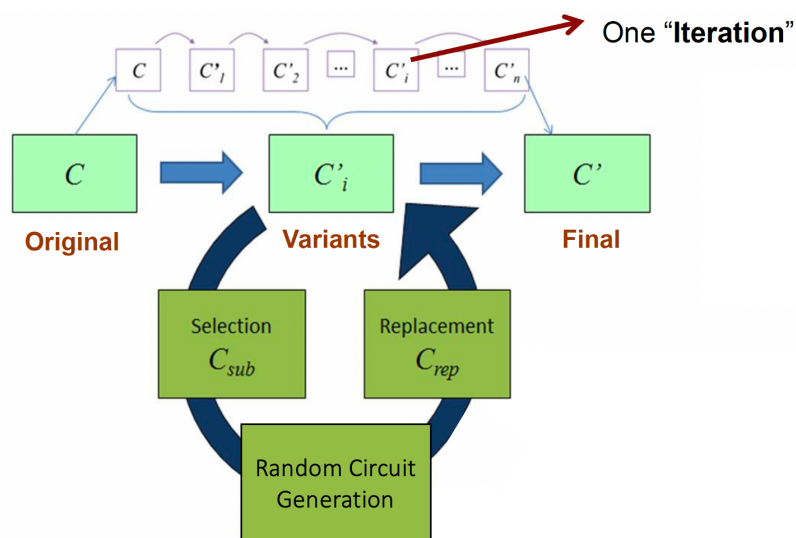


Figure 8. Iterative Selection/Replacement

The total number of gates for each S-box variant are shown in Table 3. Because the configuration for the ISR was to replace two-gate subcircuits with six-gate variants, the increase between each S-box version was expected to be at least 400 gates. However, the results shown in Table 3 reveal the gate increases to be well over that estimate. This is due to PET's Merged Signature circuit generation method, an example of which is shown in Figure 9. In the Merged Signature method, a circuit is generated for each function of the selected subcircuit and then merged to form a single circuit. If the selected subcircuit is composed of two functions, each of those may be replaced in one iteration. This accounts for the increase in gates for each S-Box being much greater than 400.

Table 3. Number of Gates for S-Box Circuit Variants

Iteration	S-Box	Number of Gates
Original	S0	1136
100	S1	1914
200	S2	2696
300	S3	3473
400	S4	4248
500	S5	5025

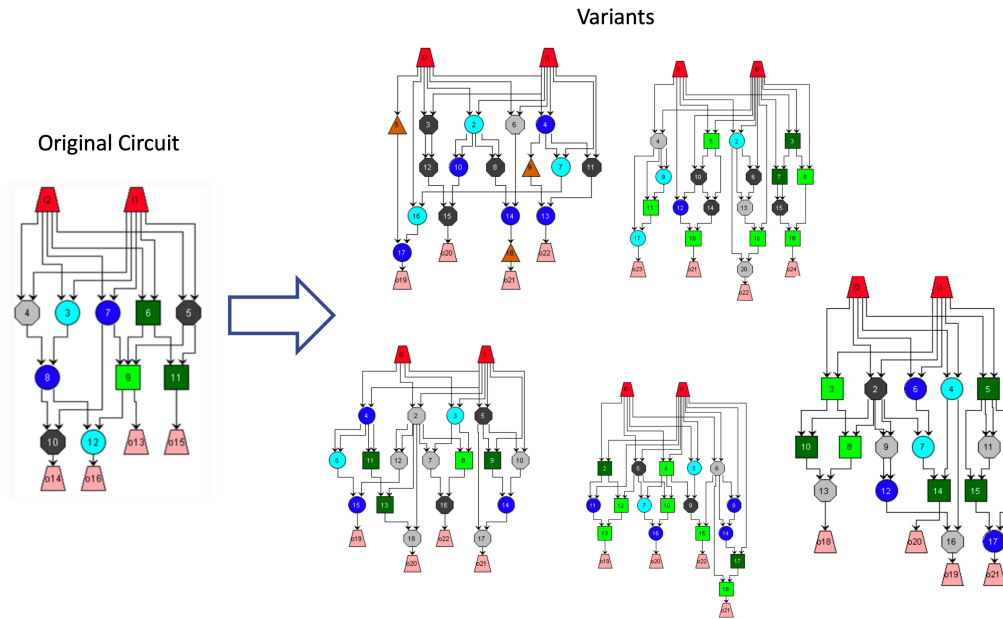


Figure 9. Merged Signature Circuit Generation. 5 Gates with a Max Fan In of 2.

4.2 Side-Channel Properties

Using PET, a VHDL source for each S-box variant was generated. Each of these VHDL S-boxes were used in their own AES core that was designed using Xilinx Vivado Design Suite 2018.1. By default, Vivado works to optimize designs for timing, power

consumption, and logic resources during synthesis and implementation [137], [138].

Therefore, these setting must be overwritten or bypassed to prevent Vivado from removing redundant logic that may have been added by PET. Custom synthesis strategies and implementation settings may be created, but logic optimization may still be automatically applied, potentially eliminating the intended effects. The best way to prevent logic from being removed is to add a DONT_TOUCH attribute to items that should not be modified. Because there is not enough available logic on the SOC to set each wire within the S-box source as DONT_TOUCH, the entire entity was specified as such instead as shown in Figure 10. This still allows Vivado to implement the designs as lookup tables (LUTs) on the device. However, the resulting Boolean logic for the LUTs vary between S-box designs. An example is given in

Table 5. Power consumption estimates were also provided by Vivado in the implemented design. The largest difference in dynamic power consumption was 1 mW. This could be due to the designs being implemented as LUTs rather than the large PET-generated circuits.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity AES_SBOX1 is
port (
    in0 : in Std_Logic;
    in1 : in Std_Logic;
    in2 : in Std_Logic;
    in3 : in Std_Logic;
    in4 : in Std_Logic;
    in5 : in Std_Logic;
    in6 : in Std_Logic;
    in7 : in Std_Logic;
    out1199 : out Std_Logic;
    out1200 : out Std_Logic;
    out1201 : out Std_Logic;
    out1202 : out Std_Logic;
    out1203 : out Std_Logic;
    out1204 : out Std_Logic;
    out1205 : out Std_Logic;
    out1206 : out Std_Logic);

attribute dont_touch : string;
attribute dont_touch of AES_SBOX1 : entity is "true";
end AES_SBOX1;
```

Figure 10. DONT_TOUCH Logic For S-box 1.

Once placed on the SOC, the execution time for each S-box version was measured. This was done using the trigger signal of the design which is high during the AES encryption. The measurements may be found in Table 4 The differences are on the order of 10 μ s but may still be observed in the EM trace. The EM traces for S-box 0 and S-box 5 may be found in Figure 11 and Figure 12, respectively.

Table 4. Execution Times for AES S-box Variants.

S-Box	Execution Time (ms)
S0	1.579
S1	1.560
S2	1.616
S3	1.631
S4	1.631
S5	1.658

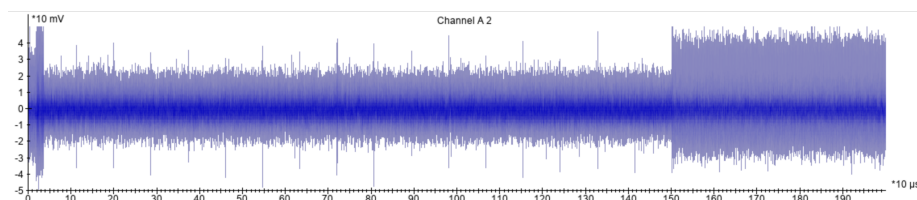


Figure 11. EM Trace of S-box 0.

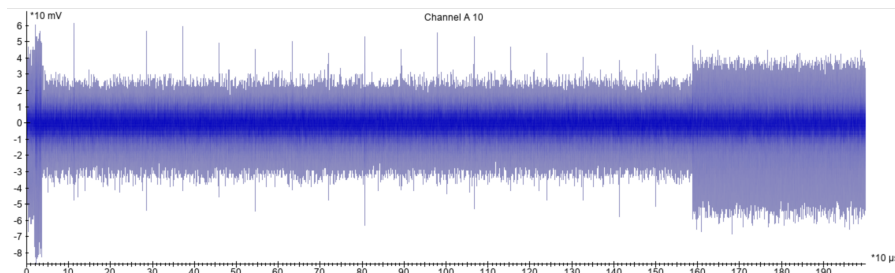


Figure 12. EM Trace of S-box 5.

Table 5. LUT Equations for S-boxes 1 and 5.

LUT	S-Box 1 Equation	S-Box 5 Equation
out1206_INST_0_i_3 (LUT6)	$O = I_0 \& !I_1 \& I_3 \& !I_4 \& I_5 + !I_0 \& !I_2 \& I_3 \& I_4 \& I_5 + !I_0 \& I_1 \& I_3 + !I_1 \& I_2 \& !I_4 + I_0 \& !I_3 \& I_4 \& I_5 + I_1 \& !I_2 \& !I_3 \& I_5 + I_0 \& !I_1 \& I_2 \& I_3 + I_1 \& !I_2 \& I_3 \& !I_5 + I_2 \& !I_3 \& !I_4 \& !I_5 + !I_0 \& !I_1 \& !I_3 \& !I_4 + I_0 \& !I_1 \& !I_2 \& I_4 \& !I_5 + !I_0 \& !I_2 \& !I_3 \& I_4 \& !I_5 + I_0 \& I_2 \& I_3 \& !I_4 \& I_5 + I_1 \& I_2 \& !I_3 \& I_4 \& I_5 + I_0 \& I_1 \& !I_3 \& !I_4 \& !I_5$	$O = I_0 \& I_1 \& !I_2 \& I_3 \& !I_5 + !I_0 \& I_1 \& I_2 \& I_3 \& !I_4 + !I_2 \& I_3 \& I_4 + !I_0 \& !I_4 \& I_5 + I_0 \& I_1 \& I_2 \& !I_3 + I_1 \& !I_3 \& I_4 \& !I_5 + !I_1 \& I_3 \& I_4 \& !I_5 + !I_0 \& !I_1 \& !I_3 \& I_5 + !I_1 \& !I_2 \& !I_3 \& !I_4 \& !I_5 + I_0 \& I_2 \& I_3 \& !I_4 \& I_5 + I_0 \& !I_1 \& I_2 \& !I_4 \& !I_5 + I_0 \& I_1 \& !I_2 \& I_4 + I_0 \& !I_2 \& I_4 \& !I_5 + !I_0 \& !I_2 \& !I_3 \& !I_4 + !I_0 \& I_1 \& I_2 \& I_3 \& I_5 + !I_0 \& I_2 \& !I_3 \& I_4 \& !I_5$
out1206_INST_0_i_4 (LUT6)	$O = I_0 \& !I_1 \& I_2 \& !I_3 \& !I_4 \& !I_5 + !I_0 \& I_1 \& I_2 \& !I_4 + !I_2 \& I_3 \& !I_4 \& !I_5 + !I_0 \& !I_2 \& !I_3 \& I_4 \& !I_5 + !I_0 \& !I_1 \& I_2 \& I_3 + I_0 \& I_1 \& I_2 \& I_4 \& !I_5 + !I_0 \& !I_3 \& !I_4 \& I_5 + I_0 \& I_1 \& !I_2 \& I_3 \& I_4 \& I_5 + I_1 \& I_2 \& !I_4 \& I_5 + I_0 \& !I_1 \& I_3 \& !I_4 \& I_5 + I_0 \& I_1 \& I_2 \& !I_3 \& I_4 + I_0 \& I_1 \& !I_3 \& I_4 \& !I_5 + !I_1 \& I_2 \& I_3 \& I_5 + !I_0 \& !I_1 \& I_2 \& I_5 + !I_0 \& I_1 \& I_2 \& I_3 \& I_5$	$O = I_0 \& !I_2 \& !I_3 \& !I_5 + !I_0 \& I_3 \& !I_4 \& !I_5 + !I_0 \& !I_1 \& I_2 \& !I_3 \& I_4 \& !I_5 + !I_0 \& !I_2 \& !I_3 \& !I_4 \& I_5 + I_0 \& I_1 \& I_2 \& I_3 \& !I_4 \& I_5 + I_0 \& !I_1 \& I_3 \& I_4 + I_0 \& I_1 \& I_4 \& !I_5 + I_1 \& !I_2 \& I_4 \& !I_5 + !I_1 \& !I_2 \& I_3 \& !I_4 + I_0 \& !I_1 \& I_2 \& I_3 \& !I_4 + I_0 \& !I_1 \& !I_2 \& I_4 + I_0 \& I_1 \& I_2 \& !I_3 \& I_4 + I_0 \& I_1 \& !I_2 \& I_3 \& I_4 + !I_0 \& I_1 \& I_2 \& !I_3 \& I_5$
out1206_INST_0_i_5 (LUT6)	$O = I_0 \& I_1 \& !I_4 \& !I_5 + !I_0 \& !I_1 \& I_2 \& !I_3 \& !I_4 + !I_0 \& I_1 \& I_2 \& I_4 \& I_5 + I_0 \& I_1 \& I_3 \& I_4 + !I_0 \& I_1 \& I_3 \& !I_4 + I_0 \& !I_1 \& I_2 \& I_3 \& !I_4 + I_0 \& !I_2 \& I_3 \& !I_4 \& I_5 + I_0 \& !I_2 \& !I_3 \& I_4 \& I_5 + !I_0 \& !I_1 \& I_3 \& I_4 \& I_5 + I_0 \& !I_1 \& !I_2 \& I_4 \& !I_5 + !I_1 \& !I_2 \& !I_3 \& !I_4 \& !I_5 + I_1 \& !I_2 \& I_3 \& !I_5 + !I_1 \& !I_2 \& !I_3 \& I_4 \& I_5$	$O = !I_1 \& I_2 \& !I_3 \& !I_4 \& !I_5 + I_0 \& I_1 \& I_2 \& !I_3 \& I_4 + !I_0 \& !I_1 \& I_3 \& I_4 + I_1 \& I_3 \& I_4 \& I_5 + !I_1 \& !I_3 \& I_4 \& I_5 + I_0 \& I_1 \& !I_2 \& I_3 \& !I_5 + I_0 \& I_1 \& !I_3 \& !I_4 \& I_5 + I_0 \& !I_1 \& I_3 \& !I_4 \& I_5 + !I_0 \& I_1 \& !I_2 \& I_3 \& !I_4 + !I_0 \& !I_1 \& !I_2 \& !I_4 \& !I_5 + I_0 \& !I_1 \& !I_2 \& I_3 \& I_5 + !I_1 \& I_2 \& I_3 \& !I_4 \& I_5 + I_0 \& I_1 \& !I_2 \& !I_4 \& I_5$
out1206_INST_0_i_6 (LUT6)	$O = !I_0 \& I_1 \& !I_2 \& !I_3 \& !I_4 \& !I_5 + !I_0 \& !I_1 \& I_2 \& !I_3 + I_0 \& I_2 \& I_3 \& !I_4 \& !I_5 + I_0 \& !I_1 \& !I_2 \& I_4 \& !I_5 + I_0 \& I_1 \& !I_2 \& !I_4 \& I_5 + I_0 \& !I_1 \& !I_2 \& !I_3 \& I_4 + !I_0 \& I_1 \& I_4 \& I_5 + !I_1 \& I_2 \& I_3 \& I_5 + !I_0 \& I_3 \& I_5 + I_0 \& I_1 \& !I_3 \& I_4 \& !I_5 + !I_0 \& I_2 \& !I_3 \& I_4 + !I_1 \& !I_2 \& !I_3 \& I_4 \& !I_5$	$O = I_1 \& I_2 \& !I_3 \& !I_4 \& !I_5 + !I_2 \& I_3 \& !I_4 \& !I_5 + !I_0 \& !I_1 \& !I_2 \& !I_3 \& I_4 \& !I_5 + I_0 \& I_1 \& !I_2 \& I_4 + I_0 \& !I_1 \& I_2 \& !I_3 \& I_4 + I_0 \& !I_2 \& I_5 + !I_0 \& I_1 \& I_2 \& !I_3 \& !I_4 + !I_0 \& !I_1 \& I_2 \& I_3 \& I_5 + !I_0 \& I_1 \& I_2 \& I_4 \& !I_5 + I_0 \& I_3 \& !I_4 \& I_5 + I_1 \& !I_2 \& I_3 \& !I_5 + !I_0 \& I_1 \& !I_3 \& !I_4 \& !I_5$

CHAPTER V

COUNTERMEASURE DESIGN

In this chapter, details of the circuit-variant moving target countermeasure design are discussed as well as the control implementation. An overview of the equipment and resources used in this research is provided.

5.1 Equipment and Resources

This research uses a Digilent ZedBoard evaluation and development platform, which features a Zynq-7000 SoC XC7Z020-CLG484-1 (Dual ARM Cortex-A9 MPCore 667MHz) complete with 85k programmable logic cells, 4.9Mb BRAM, and 512 MB DDR3 [139], [140]. Xilinx Vivado Design Suite 2018.1 was used to design and program the system. With this software, a user may synthesize designs from behavioral descriptions (e.g., VHDL code), add and configure specialized IP cores, specify placement and route (P&R) details, simulate execution, and generate then export bitstreams to the device. The AES core was designed in C using Vivado High Level Synthesis (HLS). This software allows IP behavior to be written in C, C++, or SystemC and then synthesized and output as a VHDL or Verilog-based IP source. HLS manages the incorporation of code needed to allow an IP core to serve as an AXI peripheral in order to use industry standard embedded communication protocols and interfaces. This is

useful in that it does not require any interference from the user to modify an IP source to that it may access Zynq resources (e.g., UART ports). A java-based program encryption toolkit (PET) was used to generate circuit variants as described in CHAPTER IV.

To collect power and EM traces for the analysis, a Riscure Side-Channel analysis Suite including an EM Probe station, PicoScope 3000 Series oscilloscope, and Inspector 2021.1 software was used. The probe station consists of a high-resolution EM probe and a motorized XYZ table which are integrated with the Inspector software for configuration and measurement. The coil for the EM probe has an inner area of 1 mm^2 and outer area of 2 mm^2 . The station can be setup to automatically scan the surface of the chip with a step size as small as $2.5 \text{ }\mu\text{m}$. The PicoScope 3206D oscilloscope is a USB-powered, two-channel oscilloscope with 200 MHz analog bandwidth and 1GS/s real-time sampling. The Inspector software was used to configure the equipment for trace collection, generate and send random plaintext to the target device, receive ciphertext output, store traces, perform pre-processing on traces (e.g., filtering and resampling), and statistically analyze the samples during the attacks. The target algorithm for this research is AES-128 [27]. This limits the algorithm to ten encryption rounds for a 128-bit block of plaintext.

5.2 Countermeasure Design

The AES implementation protected with our countermeasure consists of several IP blocks that were configured using Vivado. This includes the HLS AES core, Zynq Processing system, AXI GPIO, as well as AXI interconnecting IP blocks that allow for communication between the components. The connections between IP blocks are shown in the Vivado block diagram, Figure 13. The Zynq System on a Chip (SOC) features a

processing system (PS) and FPGA programmable logic (PL). The Zynq processor serves as the controller for the implementation, providing the clock and reset signals that control the other IP blocks in the design. Though the AES implementation is hardware based, the input and output are also handled by the PS side of the SOC. Interfacing applications are hosted on the PS side that are used to communicate to the device with the Inspector software (i.e., send and receive data blocks) as well as control a hardware trigger that indicates the start of an encryption run. A device block diagram is shown in Figure 14.

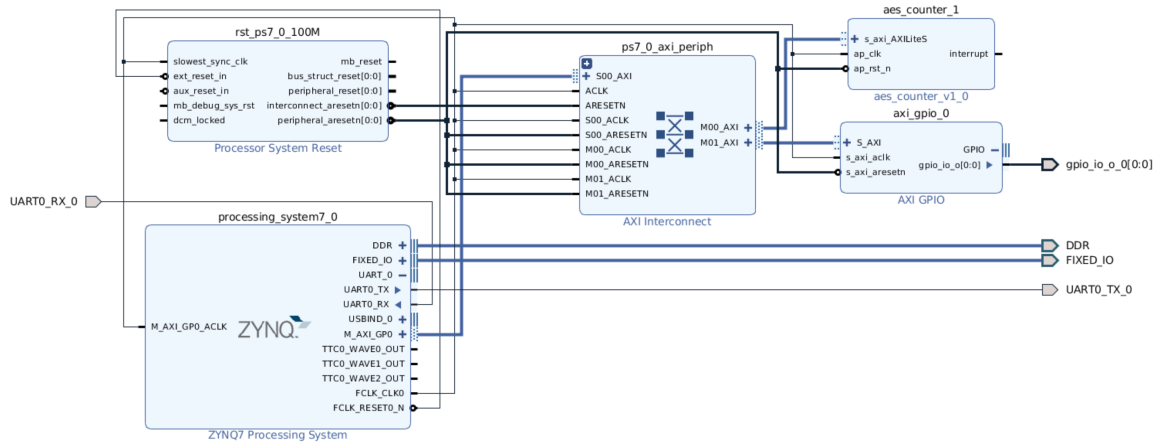


Figure 13. IP Core Block Diagram.

During each execution of the algorithm, a 128-bit block of random data is sent from the Inspector software to be encrypted by the AES scheme. When the data is received, the hardware trigger is set to high to indicate the process start. The trigger is measured with an oscilloscope that is connected to the Inspector software so that trace measurement starts at the correct time.

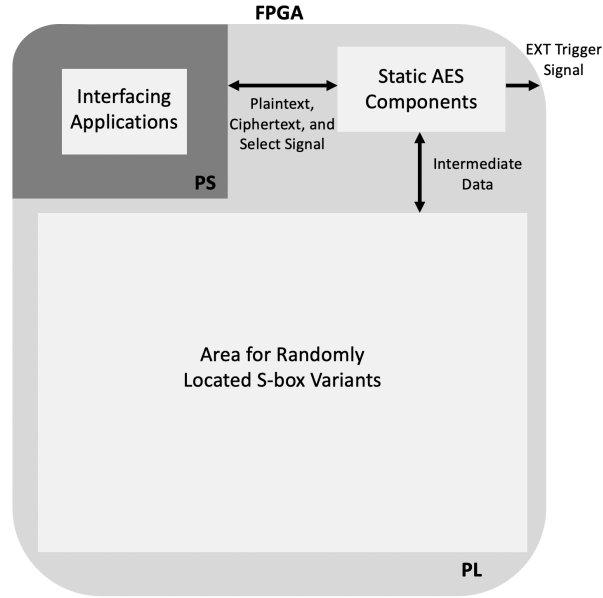


Figure 14. System Block Diagram.

The countermeasure proposed in this research aims to improve resistance to side-channel attacks by spatially and structurally randomizing an implementation of the AES encryption algorithm. Specifically, the AES S-box is the point at which the countermeasure is applied since it is of particular interest to SCA attackers. The HLS IP source for the AES core was modified to incorporate the PET-generated variants into the design. Typically, S-boxes are implemented as look-up tables in memory. That structure, however, was replaced by the six S-box variants. A multiplexor was also added to select which S-box output to use based on an input signal randomly generated by the Zynq processor.

To prevent Vivado from removing redundant S-boxes, DONT_TOUCH logic was applied to the VHDL entities as discussed in CHAPTER IV. By default, Vivado also optimizes designs for timing when performing place and route. To introduce spatial

randomization to the design, logic for each S-box variant was placed in its own P-block (i.e., placement constraint block). The remaining AES components were placed together in another P-block while all other IP blocks were placed in a P-block together. The S-box P-blocks were then mapped to the PL in a random order and placed for maximum spread between S-boxes within timing requirements. The other two P-blocks were placed to meet timing requirements by Vivado. The resulting countermeasure device layout is shown in Figure 15. It may be observed that there are 12 S-box instances rather than the six versions that were generated. This is due to the behavior of the HLS AES core which introduces some unrolling of the algorithm which results in two S-box instances. Since six versions of the S-box were added, the IP core realized an additional copy of each. Additionally, the lines in the layout between P-block represent shared nets. The red bundle net between the AES components P-block and the remaining IP P-block indicates that there are 60-200 shared nets.

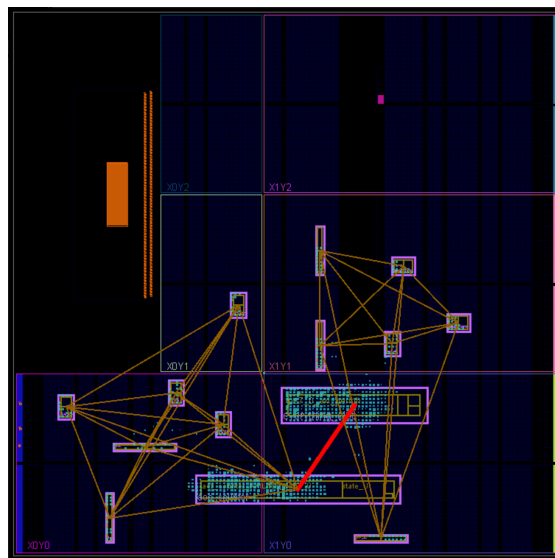


Figure 15. Countermeasure Device Layout.

5.3 Control Implementation

The control implementation utilizes the same HLS AES core as the countermeasure; however, different modifications were made to the code. Rather than introducing multiple instances of S-boxes, the memory look-up structure of the original code was replaced by S-box 0. The source code for S-box 0 is the original circuit that was used to generate five variants in PET, as discussed in Chapter CHAPTER IV. The connection of IP cores as shown in Figure 13 remains the same for the control design. For place and route, the components of the AES core were placed in one P-block while the remaining components of the design were placed in another. This was done so that the distance between the AES core and other logic could be maximized to avoid capturing noise from uninteresting processes during localized EM collection. The device layout of the control implementation is shown in Figure 16.

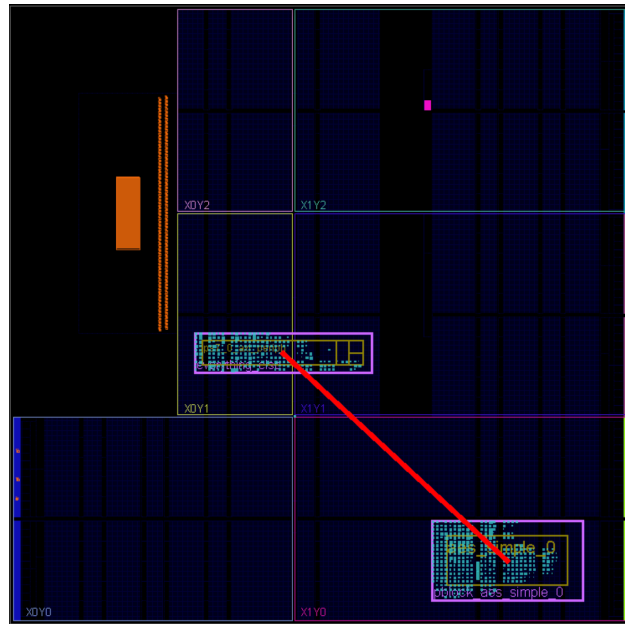


Figure 16. Control Device Layout.

CHAPTER VI

SIDE-CHANNEL RESISTANCE

In this section, the details of performing DEMA and DPA attacks on both the control and countermeasure implementations are provided. This includes collecting EM and power side-channel data, identifying usable trace sets, and performing practical evaluation of both designs using first order differential analysis. Results of these methods are also discussed.

6.1 Localized EM Analysis

For this attack, a high-resolution EM probe was connected to the PicoScope oscilloscope to capture measurements during encryption. The coil for the EM probe has an inner area of 1 mm^2 and outer area of 2 mm^2 . The location of the EM probe was controlled by an XYZ table that was configured using the Inspector software. The station can be setup to automatically scan the surface of the chip with a step size as small as $2.5 \text{ }\mu\text{m}$. These devices are represented by the EM probe station in the analysis setup shown in Figure 17. Before collecting an attack trace set, an optimal probe position must be determined. This was done by performing trace collection in an XY sweep of the chip at incremental steps.

After determining which hotspot most likely corresponded to the target hardware location, the probe was then placed at those coordinates. A set of 1000 traces each was collected for the control and countermeasure designs. Traces were collected at 250MHz which is greater than the recommended sampling rate of twice the frequency of the PL clock, 100MHz. The number of samples collected for each implementation was selected so that the execution time was within the length of the trace. For the control implementation, 500k samples were collected for a 2ms long trace and 700k samples were collected for the countermeasure design resulting in a 2.8ms long trace. A larger trace set of 2000 traces was also collected for the countermeasure implementation to use for the attack.

The Inspector software was then used to reduce noise within the trace set by filtering out unnecessary harmonics within the signal. Traces were also aligned using the first round of encryption, which was identified using an autocorrelation of the first trace. A correlation module was then used to determine the implementations' susceptibility to side-channel analysis. Finally, a first order differential analysis was performed on each implementation. This module provides a list of the best key candidates, their confidence, and their position. The first order analysis module applied in this research targeted the S-box output in the first round of AES using a Hamming Weight Model.

6.1.1 Results

The XY scan of the chip resulted in three observable hotspots at different frequency windows. Hotspots A and B, shown in Figure 18 and Figure 19, contain noisy traces that do not appear to have any artifacts of the AES encryption. However, hotspot C

shows a relatively lower amplitude pattern in the trace that occurs for the execution time of the encryption. Figure 20 shows an example from the trace set collected at hotspot C for the control implementation and Figure 23 shows one from the countermeasure implementation. The difference in execution time may be observed in each trace where the low-amplitude pattern occurs for approximately 1.55 ms in the control implementation and 2.5 ms in the countermeasure implementation. The trace set collected at hotspot C was selected as the target trace set for the attack since it was the only point at which artifacts of the encryption could be observed.

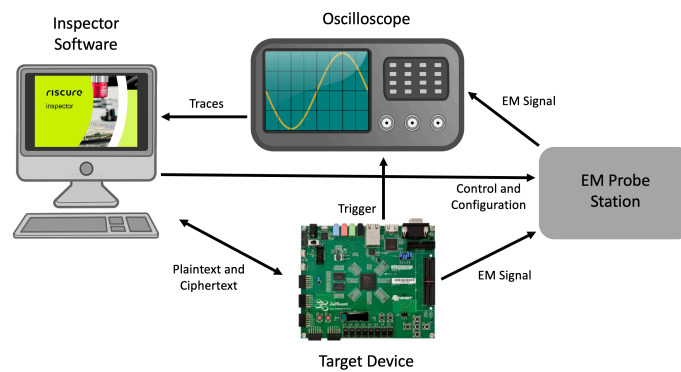


Figure 17. Electromagnetic Analysis Setup.

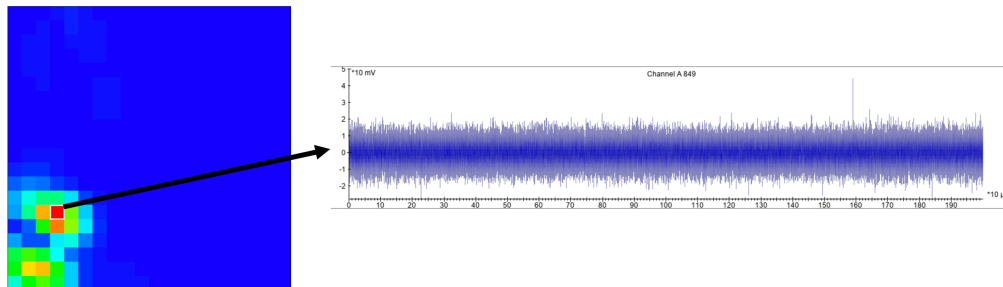


Figure 18. Control Hotspot A and Corresponding EM Trace.

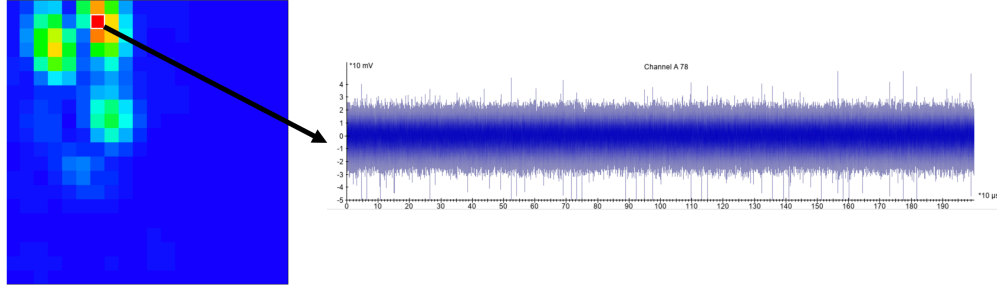


Figure 19. Control Hotspot B and Corresponding EM Trace.

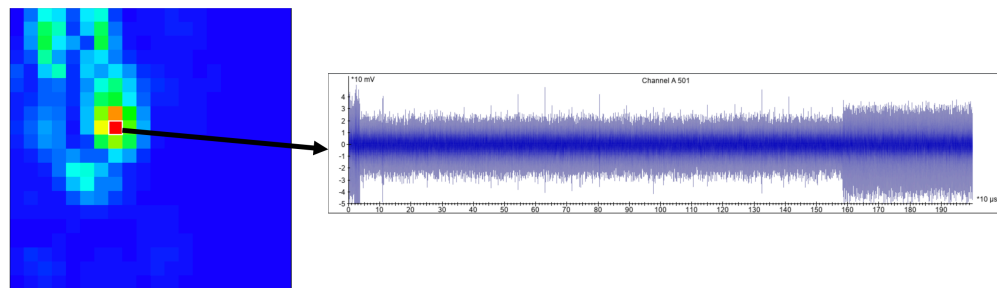


Figure 20. Control Hotspot C and Corresponding EM Trace.

Each of the observed hotspots occurred at similar coordinates for both the control and countermeasure implementations. This may be due to the Zynq processor having a higher influence on the EM field than the AES hardware. Because this is a common component in the two implementations, it could be responsible for the hotspots occurring in similar locations. Further, the additional S-box variants of the countermeasure may not have enough of an influence on the EM field to overcome the hotspots introduced by the Zynq processor. An attacker may be able to observe artifacts of AES at random locations on the chip due to the additional S-boxes, however, the XY spectral intensity of the chip would lead an attacker to collect traces at the same coordinates as the control.

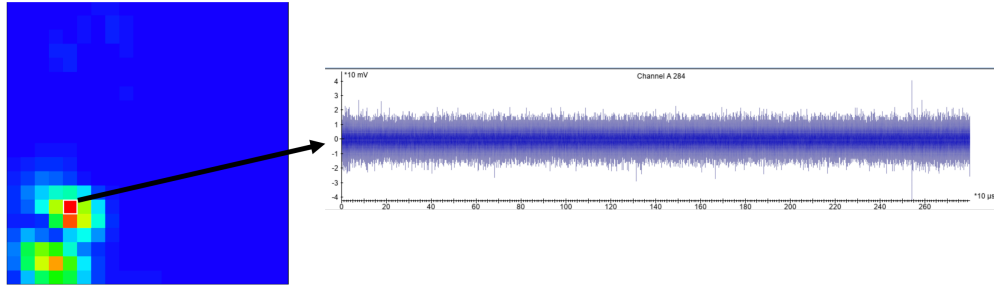


Figure 21. Countermeasure Hotspot A and Corresponding EM Trace.

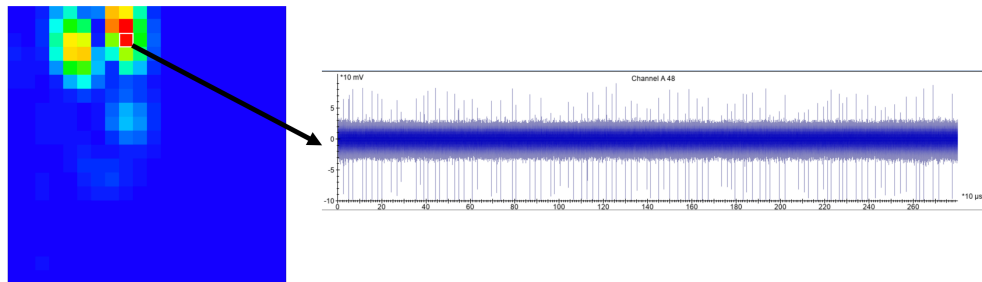


Figure 22. Countermeasure Hotspot B and Corresponding EM Trace.

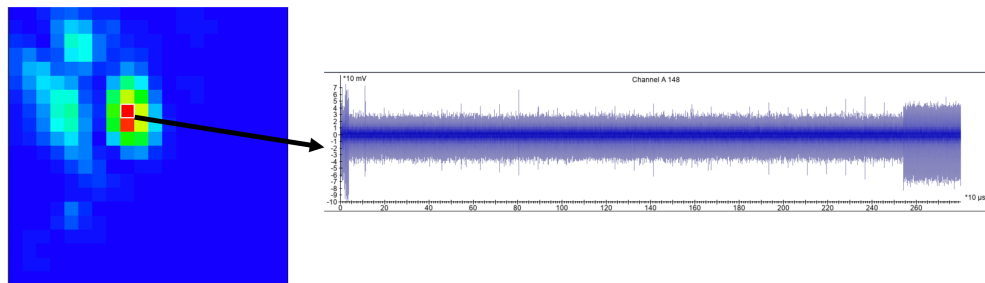


Figure 23. Countermeasure Hotspot C and Corresponding EM Trace.

Though these hotspots occurred at similar coordinates for both designs, the frequency windows at which hotspot C was varied for each implementation. For the control, C was observed between 8.789-58.838 MHz while C was observed between 65.674-99.365 MHz for the countermeasure. Though the difference in frequency

windows may not influence the side-channel leakage for the designs, it is an observable difference in behavior between the control and countermeasure implementations. With a more complex design and/or different analysis method, and attacker may be required to vary their collection methods for a similar countermeasure due to the observed frequency window difference.

The first four bits of correlation module output for both implementations are shown in Figure 24 and Figure 25. With a low-noise trace set, peaks at different times for various input bits may be observed. These results are used to determine an implementation's susceptibility to side-channel analysis and may reveal the point at which encryption occurs in a trace if it not easily observed. There is too much noise within the collected trace sets for the correlation module to be of use in an attack. However, the results do display a difference of bit-level side-channel leakage for the implementations. With a higher SNR, the correlation module would give better insight to how the countermeasure influences side-channel leakage as compared to the control.

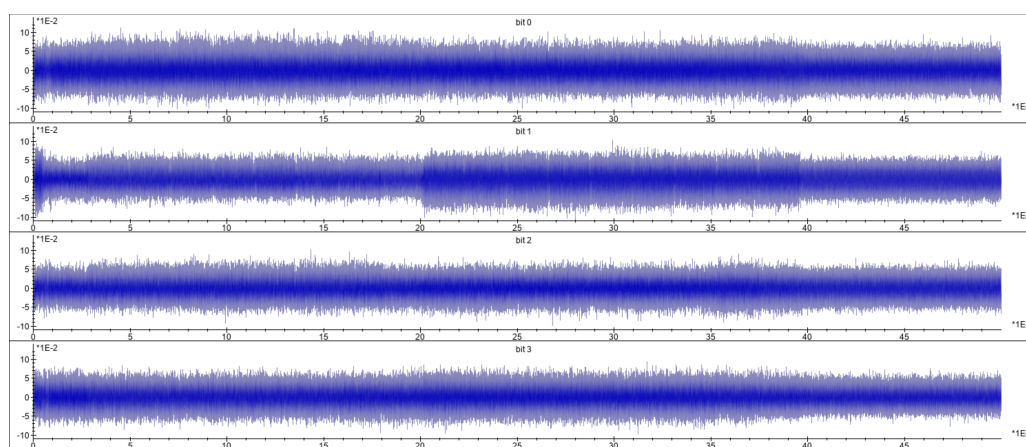


Figure 24. Control Correlation for Input Bits 0-3.

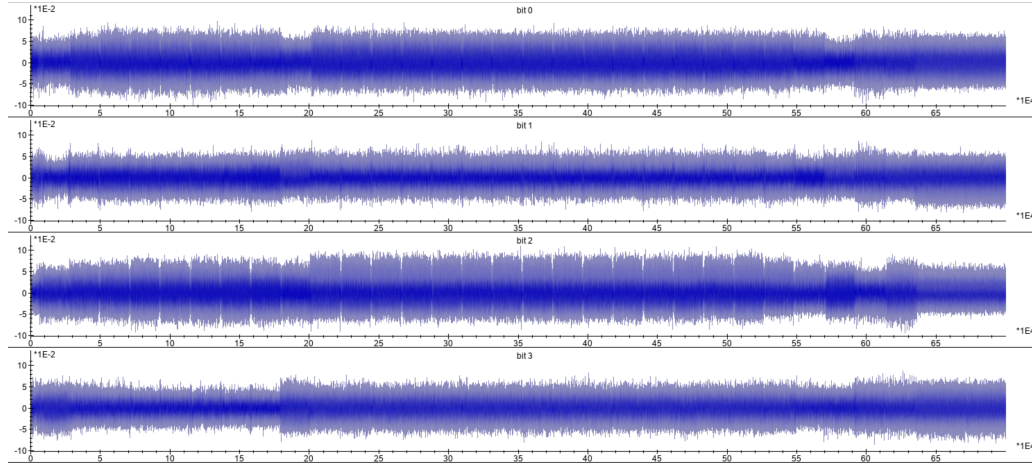


Figure 25. Countermeasure Correlation for Input Bits 0-3.

An autocorrelation of a trace from each implementation was performed to visualize repeating processes. This analysis can be useful to recognize known program structures, such as S-box substitutions. The resulting graphs showed clear artifacts of the AES encryption. Figure 26 shows the output for the control implementation in which 18 repeating squares are observed followed by a large bright square at the end of the trace. This corresponds to the nine iterations of similar rounds in AES possibly followed by the completion of the algorithm and memory writes of the results. Figure 27 shows the autocorrelation graph for the countermeasure implementation. A larger repeating square structure is observed followed by a bright square. The larger size of the repeating structure is due to the longer execution time of the countermeasure encryption. The difference in the number of squares in the structure may be due to the multiple S-box variants present in the design. The position within the traces corresponding to the first two rounds of AES was identified. The amplitude peak occurring within this window was used as the reference for static alignment of the trace sets.

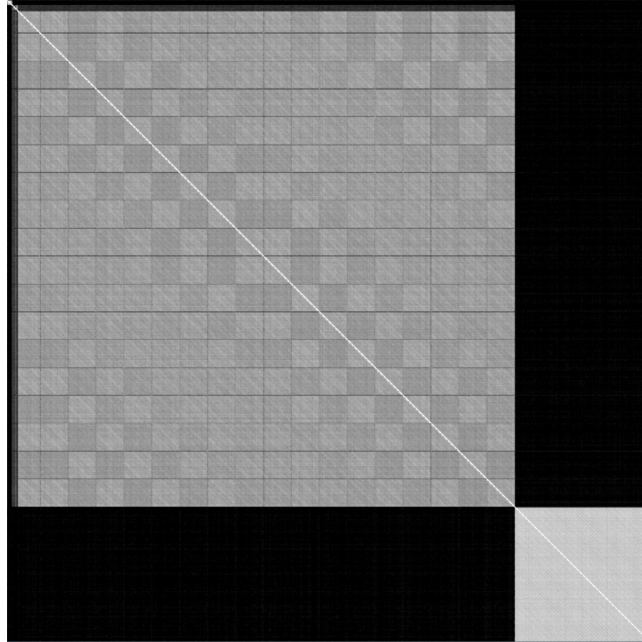


Figure 26. Control Autocorrelation Graph.

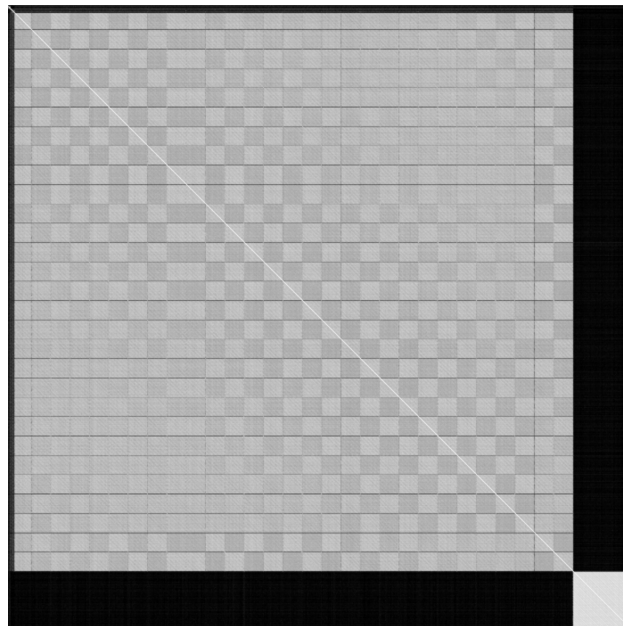


Figure 27. Coutnermeasure Autocorrelation Graph.

The samples present within the window of the first two rounds were used to perform first order analysis of the control and countermeasure implementations. The control implementation recovered six nibbles of the key within 291 traces which are shown in Table 6. Some examples of the confidence values for key candidates are shown in Figure 28. The entire output of the first order analysis may be found in Appendix A. A key byte candidate is more likely to be correct if the first ranked candidate has a much higher confidence than the other ranks. Though only one nibble for the candidates shown in Figure 28 was correct, the difference between the first and second rank for the key bytes with one correct nibble are greater than those with no correct bits.

No bytes of the key were able to be recovered from the countermeasure implementation within 1000 traces. Using a 2000 trace set, two nibbles of the key were able to be recovered, shown in Table 7. The results for those candidates along with two incorrect candidates are shown in Figure 29. The entire output of the first order analysis may be found in Appendix B. For the correct key nibbles, the confidence value difference between the first and second ranked candidates is less than many of the incorrect key candidates. Further, the overall confidence of key candidates is very small, less than 0.122 for all bytes. Therefore, the two key nibbles that were correct may have been a result of coincidence rather than side-channel leakage.

The difference in overall confidence of key candidates between the two implementations reflects a difference in leverageable side-channel leakage for the designs. Though the trace set collected from the control implementation was very noisy, nibbles of the key were still able to be recovered with clear correct key candidates. The countermeasure implementation significantly reduced the ability to extract clear key

candidates from first order differential analysis. Even with increasing usable trace set by nearly a factor of 10, the countermeasure still outperformed the control implementation in terms of SCA resistance. Though the attack could be improved by further increasing the number of traces used, it may be more beneficial to reduce the influence of the Zynq processor on the EM field. Because this component had such a great influence on the EM side-channel, the effect of S-box circuit variants may also be better observed once the Zynq processor noise is reduced. Future work is needed to determine what effect this may have on differential analysis results.

```
Best score for Round 0: Key: Column 1, Row 2 with rdm: 0.3041:
rank: 1, candidate: 94 (0x5E), confidence: 0.2934 at position: 26035
rank: 2, candidate: 249 (0xF9), confidence: 0.2867 at position: 6117
rank: 3, candidate: 209 (0xD1), confidence: 0.2798 at position: 26610
rank: 4, candidate: 62 (0x3E), confidence: 0.2795 at position: 28656
Best score for Round 0: Key: Column 2, Row 3 with rdm: 0.6103:
rank: 1, candidate: 13 (0x0D), confidence: 0.3101 at position: 25167
rank: 2, candidate: 204 (0xCC), confidence: 0.2964 at position: 26499
rank: 3, candidate: 225 (0xE1), confidence: 0.2946 at position: 42044
rank: 4, candidate: 130 (0x82), confidence: 0.2937 at position: 31988
Best score for Round 0: Key: Column 3, Row 1 with rdm: 0.2344:
rank: 1, candidate: 46 (0x2E), confidence: 0.2985 at position: 25770
rank: 2, candidate: 29 (0x1D), confidence: 0.2938 at position: 35983
rank: 3, candidate: 45 (0x2D), confidence: 0.2776 at position: 41323
rank: 4, candidate: 31 (0x1F), confidence: 0.2775 at position: 38632
Best score for Round 0: Key: Column 3, Row 2 with rdm: 0.0231:
rank: 1, candidate: 250 (0xFA), confidence: 0.2970 at position: 50299
rank: 2, candidate: 56 (0x38), confidence: 0.2964 at position: 37791
rank: 3, candidate: 158 (0x9E), confidence: 0.2952 at position: 7780
rank: 4, candidate: 61 (0x3D), confidence: 0.2930 at position: 43594
```

Figure 28. Control First Order Analysis Examples.

Best score for Round 0: Key: Column 0, Row 2 with rdm: 0.5825:
rank: 1, candidate: 254 (0xFE), confidence: 0.1075 at position: 4349
rank: 2, candidate: 202 (0xCA), confidence: 0.1037 at position: 8938
rank: 3, candidate: 220 (0xDC), confidence: 0.1030 at position: 3767
rank: 4, candidate: 122 (0x7A), confidence: 0.1029 at position: 4382
Best score for Round 0: Key: Column 2, Row 2 with rdm: 0.1465:
rank: 1, candidate: 30 (0x1E), confidence: 0.1116 at position: 6058
rank: 2, candidate: 204 (0xCC), confidence: 0.1105 at position: 9682
rank: 3, candidate: 91 (0x5B), confidence: 0.1103 at position: 10178
rank: 4, candidate: 166 (0xA6), confidence: 0.1097 at position: 6207
Best score for Round 0: Key: Column 3, Row 1 with rdm: 0.5972:
rank: 1, candidate: 50 (0x32), confidence: 0.1125 at position: 5765
rank: 2, candidate: 138 (0x8A), confidence: 0.1087 at position: 10125
rank: 3, candidate: 109 (0x6D), confidence: 0.1075 at position: 3557
rank: 4, candidate: 123 (0x7B), confidence: 0.1050 at position: 9636
Best score for Round 0: Key: Column 3, Row 2 with rdm: 0.3551:
rank: 1, candidate: 72 (0x48), confidence: 0.1132 at position: 3154
rank: 2, candidate: 213 (0xD5), confidence: 0.1107 at position: 3490
rank: 3, candidate: 50 (0x32), confidence: 0.1086 at position: 9217
rank: 4, candidate: 144 (0x90), confidence: 0.1071 at position: 8500

Figure 29. Countermeasure First Order Analysis Examples.

Table 6. Recovered Key from Control Implementation.

AES Key	0xdeadbeefbaadbeeffeedfeedcafe bab e
Recovered Key	0x2314be50f6275e5330deec0dd72efa80

Table 7. Recovered Key from Countermeasure Implementation.

AES Key	0xdeadbeefbaadbeeffeedfeedcafe ba be
Recovered Key	0xf089fed9dde4e42676491e96f932481c

6.2 Power Analysis

The ZedBoard used in this research is equipped with a 10 m Ω resistor that is in series with the power supply. The power consumption was measured across this resistor using a differential current sense probe and captured using a PicoScope oscilloscope. The setup for the power trace collection is shown in Figure 30. Similar to the EM collections, a set of 1000 traces each was collected for the control and countermeasure designs. Traces were collected at 250MHz for 500k samples and 700k samples for the control and countermeasure implementations, respectively. Several low frequency components were present in the trace sets. To reduce the noise introduced by these signals, a *XTalClear* Inspector filter was applied which blocks frequencies that do not appear at regular harmonics intervals. An autocorrelation module was also applied to the trace sets to identify any repeating processes within the first trace. Due to the noise present in the power traces, the autocorrelation module did not indicate any artifacts of AES. Consequentially, no common behavior within the waveforms could be observed and used for alignment and first order analysis.

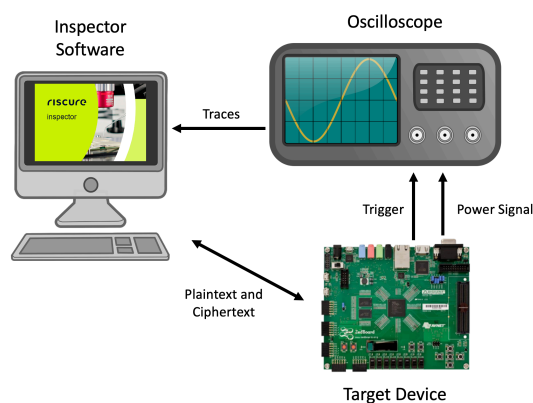


Figure 30. Power Analysis Setup.

6.2.1 Results

An example from the control implementation trace set that was collected across the current sense resistor is shown in Figure 31. After applying spectral and *XTalClear* filters to reduce noise in the traces, no identifiable artifacts of AES were observed as there were in the EM traces. An example from the resulting trace set is shown in Figure 32. An autocorrelation was still performed on the original and filtered trace sets to identify any repeating structures of AES that may not be easily observed. The resulting graphs of each are shown in Figure 33Figure 34. The traces collected from the countermeasure implementation yielded similar results.

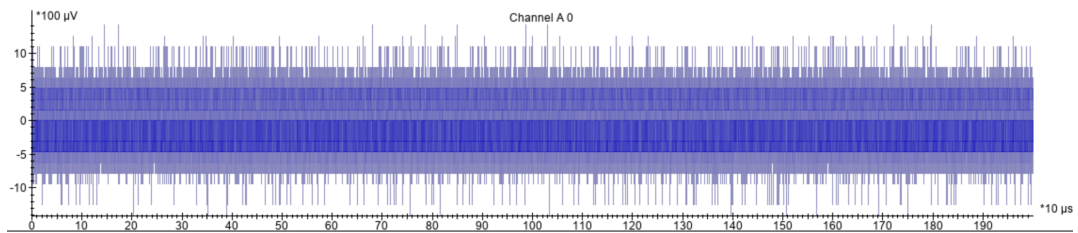


Figure 31. Control Power Trace.

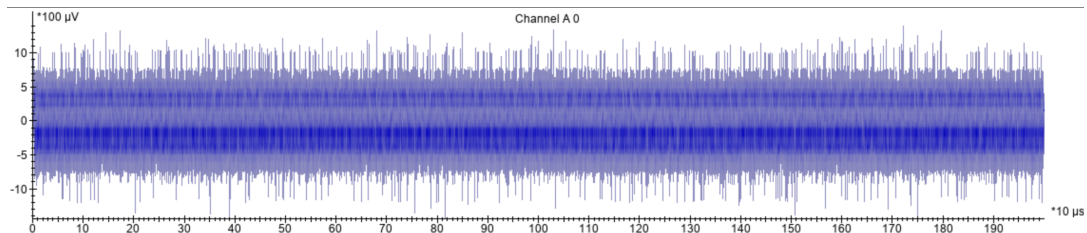


Figure 32. Filtered Control Power Trace.

The absence of AES artifacts could be due to the limitations of power analysis including being restricted to a single view of the system. When collecting EM traces, it

was possible to reduce the influence of uninteresting processes and noise by measuring at a specific location. However, because power measurements are limited to measuring across the power supply of the entire device, noise from irrelevant components and processes are included in the trace set. Because the first rounds of AES could not be identified, first order differential analysis could not be applied to the collected trace sets. While typically it is possible to improve an attack by obtaining a larger trace set, the SNR of the target process still needs to be at an exploitable level. It may be necessary to modify the control and countermeasure designs so that the Zynq processor has minimal influence on the power consumption or disable the PS entirely. Future work is needed to determine which design changes would allow a side-channel leakage assessment of both the control and countermeasure designs.

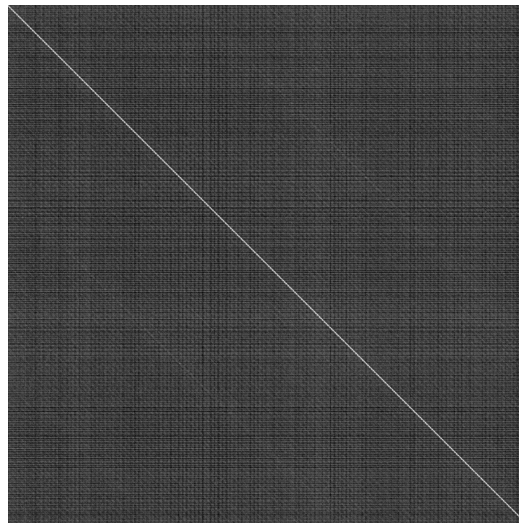


Figure 33. Control Power Autocorrelation Graph.

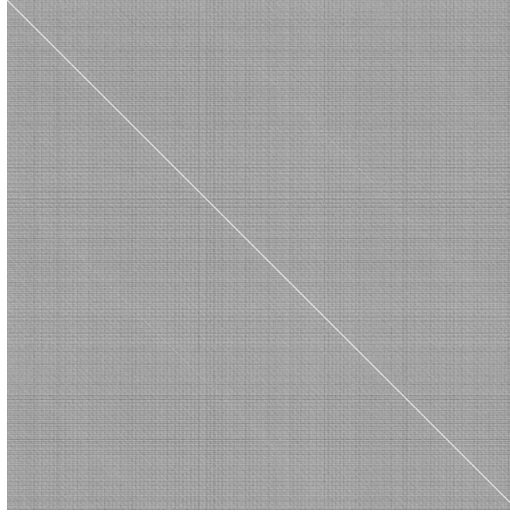


Figure 34. Filtered Control Power Autocorrelation Graph.

6.3 Performance and Cost

The effect the countermeasure had on performance may be observed directly in the EM traces that were collected. Execution time for both implementations was measuring using the trigger signal that was output by the device. A GPIO pin was set high prior to starting the encryption and low once ciphertext was obtained. This signal indicated an execution time of 1.579 ms for the control implementation and 2.598 ms for the countermeasure. Though this increase is significant, different circuit variants may yield different result. Each of the variants used in the countermeasure design increase in size from S-box 0 to 5; however, a significant size increase may be avoided by adjusting parameters for generation in PET. While reducing the overall size of circuit replacements would result in less timing overhead for the countermeasure, it may also result in less power consumption diversity which is needed to thwart DPA attacks. Future work is

needed to determine the relationship between circuit variant size and side-channel leakage.

Table 8. Utilization of Control Implementation.

Site Type	Used	Available	Util%
Slice LUTs	1736	53200	3.26
LUT as Logic	1674	53200	3.15
LUT as Memory	62	17400	0.36
Slice Registers	1845	106400	1.73
F7 Muxes	118	26600	0.44
F8 Muxes	40	13300	0.30

Table 9. Utilization of Countermeasure Implementation.

Site Type	Used	Available	Util%
Slice LUTs	2381	53200	4.48
LUT as Logic	2319	53200	4.36
LUT as Memory	62	17400	0.36
Slice Registers	1849	106400	1.74
F7 Muxes	448	26600	1.68
F8 Muxes	209	13300	1.57

A summary of the utilization statistics is provided in Table 8 for the control implementation and Table 9 for the countermeasure. For the control implementation, 68.28% of the logic LUTs were utilized by the AES core while 77.12% of the logic LUTs were used for the countermeasure AES core. Overall, the countermeasure design resulted in a 37.15% increase in LUTs compared to the control. This may be due to the use of the DONT_TOUCH attributes in the multiple S-box instances of the countermeasure. This logic prevented Vivado from removing redundant logic which may have applied to the S-box variants since they are functionally equivalent. Adding DONT_TOUCH attributes to other parts of the S-box source code may yield different utilization, performance, and

side-channel behavior results. However, future research into these effects may be limited by the available resources on the ZedBoard. For example, DONT_TOUCH attributes cannot be applied to each of the wires of the S-box circuit variants without exceeding the number of available slice LUTs. Future work is needed to determine what influence limiting optimization of other components has on resource usage and side-channel behavior as well as the integrity of PET-generated variants.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

With the emergence of side-channel attacks, traditional methods of reducing secret key access may not be sufficient for protecting an encryption scheme. By observing behavior such as timing, power consumption, and EM radiation, an attacker may be able to correlate measurements to secret key values. Therefore, it is important to consider leakage characteristics of designs when working with cryptographic algorithms. Countermeasures for side-channel analysis may be used to prevent such attacks; however, it is often necessary to implement combinations of protections to provide sufficient resistance.

The similarities between power and EM side-channels may allow an attacker who is in possession of a device the ability to pivot between attacks. Not only can traces for power and EM be collected with simple measurement setups, but methods for analysis are also very similar for both. Therefore, not only is it of particular interest to a designer to combine countermeasures to address the shortcomings of individual protection methods, but to protect against multiple types of side-channel attacks.

This research proposes a method of circuit-variant moving target defense for power and EM side-channel attacks. The goal of this countermeasure was to reduce an attacker's usable trace set by randomizing the location and circuit structure of the AES S-

box. The side-channel impact of the proposed design was studied via four research objectives; 1) Determining if PET-generated circuit variants have an observable influence on side-channel behavior, 2) Investigating how randomly placed S-box variants are represented in the EM spectralintensity graph of a SOC, 3) Assessing trace sets for usability for side-channel analysis, 4) Performing differential first order analysis attacks on usable trace sets.

The side-channel behavior of the PET-generated variants was studied by implementing control AES designs each with one version of the S-box circuit. Because there were not enough LUT slices available on the ZedBoard to set each individual wire in the S-box source as DONT_TOUCH, only the entity for the source had the attribute applied. A difference in execution time was visible in the EM trace sets for each of the variants, increasing as the size of the circuit grew. This result indicated that PET-generated variants may be used to randomize the power and timing characteristics of a design. Future work is needed to determine how altering the components excluded from optimization (i.e., DONT_TOUCH attributes) effects the side-channel behavior. Further, a variation in timing is not necessarily reflective of the leakage characteristics of the variants. Future work is needed to determine how PET parameters translate into side-channel leakage including fan-in sizes, redundancy, and wire lengths.

When comparing the spectralintensity graphs of the control and countermeasure implementations, hotspots could be observed in similar locations. This may be due to the common components between the implementations having the most influence on the EM side-channel (e.g., the Zynq Processor). The hotspot corresponding to the target hardware was observed within a different frequency window for the countermeasure. This may be a

result of the multiple S-box circuit variants present in the design, influencing the frequency contents of the captured signal.

The usability of the collected power and EM trace sets was determined by the inclusion of AES artifacts such as repeated round structures visible within the traces. While the EM traces clearly reflected the execution time of the algorithm as well as the rounds of encryption, no clear AES artifacts could be observed via the power side-channel. This may be due to the limitations of power measurements including only being limited to one view of the system. With the EM trace collection, the probe could be placed over the location that reflected a potentially usable trace set. Contrarily, power measurements can only be taken across the power or ground supply for the board across the current sense resistor. This limits the options for noise reduction at the time of collection since all components on the device contribute to the behavior observed. It is likely that the inclusion of the Zynq processor was a large contributor to the noise in the power trace. Future work should focus on improving the SNR of the traces especially the power signal.

The absence of AES artifacts in the power trace prevented a target window for a DPA attack to be identified. However, because the rounds of AES could be observed in the EM signal for both the control and countermeasure implementations, first order analysis could be applied. Though the entire key could not be recovered from either implementation, the confidence values for the key candidates revealed a difference in the strength of the DEMA attacks on the control and countermeasure designs. While clear correct key candidates could be observed in the control implementation within 291 traces, the confidence of key candidates was significantly reduced in the countermeasure even

up to 2000 traces. Further, first ranked values showed little difference in confidence over lower ranked candidates, indicating that what bits of the key that may be recovered are not supported with great confidence. This is reflective of a weak attack and supports that the introduction of randomly located S-boxes in the countermeasure increases the number of required traces to disclose the secret key.

Another area of future work is to implement a DPR version of the countermeasure. For this concept, only one S-box variant is connected in the logic at a time and replaced intermittently using a partial bitstream containing a different version. The decision to implement a DLR scheme rather than a DPR scheme for this research was to avoid the potential noise from reconfiguration logic. Additionally, it is possible that the reconfiguration logic may be of aid to an attacker in that any observable artifacts of triggering the PRC would indicate the point at which the device has been changed. This may prompt the attacker to discard measurements until another trigger is detected or the triggering artifact may be used by the attacker to parse the trace set for useful measurements. However, if the functions of the PS used in this research could be replaced by other logic in a DPR scheme, it may improve the SNR of the side-channel signals. Further, larger circuit variants may be used since they would not need to share resources with collocated S-box version. Additional research is needed to determine which scheme results in a higher SNR as well as what effect larger circuit structures would have on the overall performance and leakage of the design.

REFERENCES

- [1] T. Wollinger and C. Paar, “How secure are FPGAs in cryptographic applications?,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2778, pp. 91–100, 2003, doi: 10.1007/978-3-540-45234-8_10.
- [2] C. Meadows, “What makes a cryptographic protocol secure? the evolution of requirements specification in formal cryptographic protocol analysis,” in *European Symposium on Programming*, 2003, pp. 10–21.
- [3] P. C. Kocher, “Cryptanalysis of Diffie-Hellman, RSA, DSS, and other systems using timing attacks,” 1995.
- [4] P. Kocher, J. Jaffe, B. Jun, and T. Caddy, “Differential power analysis,” in *Annual International Cryptology Conference*, 1999, pp. 388–397, doi: 10.1007/978-1-4419-5906-5_196.
- [5] J.-J. Quisquater and D. Samyde, “Electromagnetic analysis (ema): Measures and counter-measures for smart cards,” in *International Conference on Research in Smart Cards*, 2001, pp. 200–210.
- [6] J. Kelsey, B. Schneier, D. Wagner, and C. Hall, “Side channel cryptanalysis of

- product ciphers,” in *European Symposium on Research in Computer Security*, 1998, pp. 97–110.
- [7] C. Clavier, J.-S. Coron, and N. Dabbous, “Differential power analysis in the presence of hardware countermeasures,” *CHES 2000*, vol. 1965 LNCS, pp. 252–263, 2000, doi: 10.1007/3-540-44499-8-20.
 - [8] J.-S. Coron and I. Kizhvatov, “An efficient method for random delay generation in embedded software,” *CHES 2009*, vol. 5747 LNCS, pp. 156–170, 2009, doi: 10.1007/978-3-642-04138-9_12.
 - [9] J.-S. Coron and I. Kizhvatov, “Analysis and improvement of the random delay countermeasure of CHES 2009,” *CHES 2010*, vol. 6225 LNCS, pp. 95–109, 2010, doi: 10.1007/978-3-642-15031-9_7.
 - [10] A. Bogdanov, M. Rivain, P. S. Vejre, and J. Wang, “Higher-Order DCA against Standard Side-Channel Countermeasures,” *COSADE 2019*, vol. 11421 LNCS, pp. 118–141, 2019, doi: 10.1007/978-3-030-16350-1_8.
 - [11] T. Korak, T. Plos, and M. Hutter, “Attacking an AES-enabled NFC tag: Implications from design to a real-world scenario,” *COSADE 2012*, vol. 7275 LNCS, pp. 17–32, 2012, doi: 10.1007/978-3-642-29912-4_2.
 - [12] A. Moradi, O. Mischke, and C. Paar, “Practical evaluation of DPA countermeasures on reconfigurable hardware,” *HOST 2011*, pp. 154–160, 2011, doi: 10.1109/HST.2011.5955014.

- [13] M. Rivain, E. Prouff, and J. Doget, “Higher-order masking and shuffling for software implementations of block ciphers,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, vol. 5747 LNCS, pp. 171–188, doi: 10.1007/978-3-642-04138-9_13.
- [14] M. Rivain and E. Prouff, “Provably secure higher-order masking of AES,” *CHES 2010*, vol. 6225 LNCS, pp. 413–427, 2010, doi: 10.1007/978-3-642-15031-9_28.
- [15] J.-S. Coron and L. Goubin, “On Boolean and Arithmetic Masking against Differential Power Analysis,” *CHES 2000*, pp. 231–237, 2000, doi: 10.1007/3-540-44499-8_18.
- [16] N. E. C. Akkaya, B. Erbagci, R. Carley, and K. Mai, “A DPA-resistant self-timed three-phase dual-rail pre-charge logic family,” *HOST 2015*, pp. 112–117, 2015, doi: 10.1109/HST.2015.7140248.
- [17] W. He, E. De La Torre, and T. Riesgo, “An interleaved EPE-immune PA-DPL structure for resisting concentrated EM side channel attacks on FPGA implementation,” *COSADE 2012*, vol. 7275 LNCS, pp. 39–53, 2012, doi: 10.1007/978-3-642-29912-4_4.
- [18] T. Popp and S. Mangard, “Masked dual-rail pre-charge logic: DPA-resistance without routing constraints,” *CHES 2005*, vol. 3659, pp. 172–186, 2005, doi: 10.1007/11545262_13.

- [19] D. Sokolov, J. Murphy, A. Bystrov, and A. Yakovlev, “Improving the security of dual-rail circuits,” *CHES 2004*, vol. 3156, pp. 282–297, 2004, doi: 10.1007/978-3-540-28632-5_21.
- [20] F.-X. Standaert, O. Pereira, Y. Yu, J.-J. Quisquater, M. Yung, and E. Oswald, “Leakage resilient cryptography in practice,” in *Towards Hardware-Intrinsic Security*, Springer, 2010, pp. 99–134.
- [21] T. Güneysu and A. Moradi, “Generic Side-Channel Countermeasures for Reconfigurable Devices,” *CHES 2011*, pp. 33–48, 2011.
- [22] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, “The EM side—channel (s),” in *International workshop on cryptographic hardware and embedded systems*, 2002, pp. 29–45.
- [23] R. Specht, V. Immler, F. Unterstein, J. Heyszl, and G. Sig, “Dividing the threshold: Multi-probe localized em analysis on threshold implementations,” in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2018, pp. 33–40.
- [24] V. Immler, R. Specht, and F. Unterstein, “Your rails cannot hide from localized EM: how dual-rail logic fails on FPGAs,” in *International Conference on Cryptographic Hardware and Embedded Systems*, 2017, pp. 403–424.
- [25] J. Fan, X. Guo, E. De Mulder, P. Schaumont, B. Preneel, and I. Verbauwhede, “State-of-the-art of secure ECC implementations: A survey on known side-channel

- attacks and countermeasures,” *HOST 2010*, pp. 76–87, 2010, doi: 10.1109/HST.2010.5513110.
- [26] J. Fan and I. Verbauwhede, “An updated survey on secure ECC implementations: Attacks, countermeasures and cost,” *Other*, vol. 6805 LNCS, pp. 265–282, 2012, doi: 10.1007/978-3-642-28368-0_18.
- [27] N.-F. Standard, “Announcing the advanced encryption standard (AES),” *Fed. Inf. Process. Stand. Publ.*, vol. 197, no. 1–51, p. 3, 2001.
- [28] J. Daemen and V. Rijmen, “The block cipher Rijndael,” in *International Conference on Smart Card Research and Advanced Applications*, 1998, pp. 277–284.
- [29] J. A. Ambrose, S. Parameswaran, and A. Ignjatovic, “MUTE-AES: A multiprocessor architecture to prevent power analysis based side channel attack of the AES algorithm,” in *2008 IEEE/ACM International Conference on Computer-Aided Design*, 2008, pp. 678–684.
- [30] S. Mangard, E. Oswald, T. Popp, and T. P. Stefan Mangard Elisabeth Oswald, *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*, vol. 31. 2007.
- [31] E. Oswald, S. Mangard, N. Pramstaller, and V. Rijmen, “A side-channel analysis resistant description of the AES S-box,” in *International workshop on fast software encryption*, 2005, pp. 413–423.

- [32] S. Mangard and K. Schramm, “Pinpointing the side-channel leakage of masked AES hardware implementations,” *CHES 2006*, vol. 4249 LNCS, pp. 76–90, 2006, doi: 10.1007/11894063_7.
- [33] A. W. Fritzke, “Obfuscating against side-channel power analysis using hiding techniques for aes,” 2012.
- [34] P. Kocher, J. Jaffe, B. Jun, and others, “Introduction to differential power analysis and related attacks,” vol. 4, no. 1, pp. 64–75, 1998.
- [35] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, “Introduction to differential power analysis,” *J. Cryptogr. Eng.*, vol. 1, no. 1, pp. 5–27, 2011, doi: 10.1007/s13389-011-0006-y.
- [36] P. Kocher, J. Jaffe, B. Jun, and T. Caddy, “Differential power analysis,” in *Annual International Cryptology Conference*, 1999, pp. 388–397, doi: 10.1007/978-1-4419-5906-5_196.
- [37] E. Biham and A. Shamir, “Power analysis of the key scheduling of the AES candidates,” in *Proceedings of the second AES Candidate Conference*, 1999, pp. 115–121.
- [38] C. Clavier, D. Marion, and A. Wurcker, “Simple power analysis on aes key expansion revisited,” *CHES 2014*, vol. 8731, pp. 279–297, 2014, doi: 10.1007/978-3-662-44709-3_16.
- [39] R. Mayer-Sommer, “Smartly analyzing the simplicity and the power of simple

- power analysis on smartcards,” in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2000, pp. 78–92.
- [40] T. S. Messerges, “Using second-order power analysis to attack DPA resistant software,” in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2000, pp. 238–251.
 - [41] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, “Investigations of Power Analysis Attacks on Smartcards.,” *Smartcard*, vol. 99, pp. 151–161, 1999.
 - [42] S. Mangard, “A simple power-analysis (SPA) attack on implementations of the AES key expansion,” in *International Conference on Information Security and Cryptology*, 2002, pp. 343–358.
 - [43] J.-J. J. Quisquater and D. Samyde, “Electromagnetic analysis (Ema): Measures and countermeasures for smart cards,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2140, pp. 200–210, 2001, doi: 10.1007/3-540-45418-7_17.
 - [44] K. Gandolfi, C. Mourtel, and F. Olivier, “Electromagnetic analysis: Concrete results,” in *International workshop on cryptographic hardware and embedded systems*, 2001, pp. 251–261.
 - [45] F. DeBeer, M. Witteman, B. Gedrojc, and Y. Sheng, “Practical electromagnetic analysis,” 2011.
 - [46] W. Aerts, E. De Mulder, B. Preneel, G. A. E. Vandenbosch, and I. Verbauwhede,

- “Matching shielded loops for cryptographic analysis,” in *1st European Conference on Antennas and Propagation-EuCAP 2006*, 2006, pp. 1–6.
- [47] J. Heyszl, D. Merli, B. Heinz, F. De Santis, and G. Sigl, “Strengths and limitations of high-resolution electromagnetic field measurements for side-channel analysis,” in *International Conference on Smart Card Research and Advanced Applications*, 2012, pp. 248–262.
- [48] G. Li, V. Iyer, and M. Orshansky, “Securing AES against Localized em Attacks through Spatial Randomization of Dataflow,” *HOST 2019*, pp. 191–197, 2019, doi: 10.1109/HST.2019.8741026.
- [49] B. Hettwer, J. Petersen, S. Gehrler, H. Neumann, and T. Güneysu, “Securing cryptographic circuits by exploiting implementation diversity and partial reconfiguration on FPGAs,” in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019, pp. 260–263.
- [50] F. X. Standaert and C. Archambeau, “Using subspace-based template attacks to compare and combine power and electromagnetic information leakages,” *CHES 2008*, vol. 5154 LNCS, pp. 411–425, 2008, doi: 10.1007/978-3-540-85053-3_26.
- [51] S. Tiran, S. Ordas, Y. Teglia, M. Agoyan, and P. Maurine, “A frequency leakage model for SCA,” *HOST 2014*, no. 3, pp. 97–100, 2014, doi: 10.1109/HST.2014.6855577.
- [52] A. Moradi and O. Mischke, “How far should theory be from practice? Evaluation

of a countermeasure,” *CHES 2012*, vol. 7428 LNCS, pp. 92–106, 2012, doi: 10.1007/978-3-642-33027-8_6.

- [53] Y. Ishai, A. Sahai, and D. Wagner, “Private circuits: Securing hardware against probing attacks,” in *Annual International Cryptology Conference*, 2003, pp. 463–481.
- [54] K. Schramm and C. Paar, “Higher order masking of the AES,” in *Cryptographers’ track at the RSA conference*, 2006, pp. 208–225.
- [55] M. Rivain, E. Dottax, and E. Prouff, “Block ciphers implementations provably secure against second order side channel analysis,” in *International Workshop on Fast Software Encryption*, 2008, pp. 127–143.
- [56] M. Rivain and E. Prouff, “Provably secure higher-order masking of AES,” in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2010, pp. 413–427.
- [57] E. Prouff and T. Roche, “Higher-order glitches free implementation of the AES using secure multi-party computation protocols,” in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2011, pp. 63–78.
- [58] S. Mangard, T. Popp, and B. M. Gammel, “Side-channel leakage of masked CMOS gates,” in *Cryptographers’ Track at the RSA Conference*, 2005, pp. 351–365.
- [59] S. Mangard, N. Pramstaller, and E. Oswald, “Successfully attacking masked AES

- hardware implementations,” in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2005, pp. 157–171.
- [60] S. Mangard and K. Schramm, “Pinpointing the side-channel leakage of masked AES hardware implementations,” in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2006, pp. 76–90.
 - [61] H. Gross and S. Mangard, “Reconciling $d + 1$ masking in hardware and software,” *CHES 2017*, vol. 10529 LNCS, pp. 115–136, 2017, doi: 10.1007/978-3-319-66787-4_6.
 - [62] J.-S. Coron, F. Rondepierre, and R. Zeitoun, “High Order Masking of Look-up Tables with Common Shares,” *tCHES 2018*, vol. 2018, no. 1, pp. 40–72, 2018, doi: 10.13154/TCHES.V2018.I1.40-72.
 - [63] S. Faust, V. Grosso, S. M. Del Pozo, C. Paglialonga, and F.-X. Standaert, “Composable Masking Schemes in the Presence of Physical Defaults & the Robust Probing Model,” *tCHES 2018*, vol. 2018, Issu, no. 3, pp. 89–120, 2018, doi: 10.13154/tches.v2018.i3.89-120.
 - [64] F. Wegener, C. Baiker, and A. Moradi, “Shuffle and Mix: On the Diffusion of Randomness in Threshold Implementations of Keccak,” *COSADE 2019*, vol. 11421 LNCS, pp. 270–284, 2019, doi: 10.1007/978-3-030-16350-1_15.
 - [65] T. Güneysu and A. Moradi, “Generic Side-Channel Countermeasures for Reconfigurable Devices,” *CHES 2011*, pp. 33–48, 2011.

- [66] N. Mentens, B. Gierlichs, and I. Verbauwhede, “Power and fault analysis resistance in hardware through dynamic reconfiguration,” in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2008, pp. 346–362.
- [67] C. Suresh, R. R. Josyula, and P. Rohatgi, “Template Attacks,” *CHES 2002*, pp. 68–82, 2002, doi: 10.1109/FDTC.2014.17.
- [68] T. S. A. Satoh, T. Sugawara, N. Homma, and T. Aoki, “Development of side-channel attack standard evaluation environment,” in *2009 European Conference on Circuit Theory and Design*, 2009, pp. 403–408.
- [69] P. Sasdrich, A. Moradi, O. Mischke, and T. Guneyasu, “Achieving side-channel protection with dynamic logic reconfiguration on modern FPGAs,” *HOST 2015*, pp. 130–136, 2015, doi: 10.1109/HST.2015.7140251.
- [70] T. De Cnudde, M. Ender, and A. Moradi, “Hardware masking, revisited,” *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 2, pp. 123–148, 2018.
- [71] D. Das, S. Maity, S. Bin Nasir, S. Ghosh, A. Raychowdhury, and S. Sen, “High efficiency power side-channel attack immunity using noise injection in attenuated signature domain,” in *2017 IEEE HOST*, 2017, pp. 62–67.
- [72] D. Sokolov, J. Murphy, A. Bystrov, and A. Yakovlev, “Improving the Security of Dual-Rail Circuits,” 2004, vol. 3156, pp. 282–297, doi: 10.1007/978-3-540-28632-5_21.

- [73] J.-S. Coron, “Resistance against differential power analysis for elliptic curve cryptosystems,” in *International workshop on cryptographic hardware and embedded systems*, 1999, pp. 292–302.
- [74] E. Käsper and P. Schwabe, “Faster and Timing-Attack Resistant AES-GCM,” 2009, vol. 5747, pp. 1–17, doi: 10.1007/978-3-642-04138-9_1.
- [75] A. Bogdanov, M. Rivain, P. S. Vejre, and J. Wang, “Higher-order DCA against standard side-channel countermeasures,” in *International Workshop on Constructive Side-Channel Analysis and Secure Design*, 2019, pp. 118–141.
- [76] T. Korak, T. Plos, and M. Hutter, “Attacking an AES-enabled NFC tag: Implications from design to a real-world scenario,” in *International Workshop on Constructive Side-Channel Analysis and Secure Design*, 2012, pp. 17–32.
- [77] C. Clavier, J.-S. Coron, and N. Dabbous, “Differential power analysis in the presence of hardware countermeasures,” in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2000, pp. 252–263.
- [78] P. Sasdrich, A. Moradi, O. Mischke, and T. Güneysu, “Achieving side-channel protection with dynamic logic reconfiguration on modern FPGAs,” in *2015 IEEE HOST*, 2015, pp. 130–136.
- [79] J. Balasch, B. Gierlichs, O. Reparaz, and I. Verbauwhede, “DPA, Bitslicing and Masking at 1 GHz,” in *Cryptographic Hardware and Embedded Systems -- CHES 2015*, 2015, pp. 599–619.

- [80] L. Goubin and J. Patarin, “DES and differential power analysis the ‘Duplication’ method,” in *International Workshop on Cryptographic Hardware and Embedded Systems*, 1999, pp. 158–172.
- [81] J.-S. Coron and L. Goubin, “On Boolean and Arithmetic Masking against Differential Power Analysis,” in *Cryptographic Hardware and Embedded Systems --- CHES 2000*, 2000, pp. 231–237.
- [82] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*. Berlin, Heidelberg: Springer-Verlag, 2007.
- [83] M.-L. Akkar and C. Giraud, “An Implementation of DES and AES, Secure against Some Attacks,” 2001, pp. 309–318, doi: 10.1007/3-540-44709-1_26.
- [84] L. De Meyer, O. Reparaz, and B. Bilgin, “Multiplicative Masking for AES in Hardware,” *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 3, pp. 431–468, 2018, doi: 10.13154/tches.v2018.i3.431-468.
- [85] M. Rivain, E. Prouff, and J. Doget, “Higher-order masking and shuffling for software implementations of block ciphers,” in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2009, pp. 171–188.
- [86] W. Fischer and B. M. Gammel, “Masking at gate level in the presence of glitches,” in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2005, pp. 187–200.

- [87] A. Moradi and O. Mischke, “Glitch-free implementation of masking in modern FPGAs,” in *2012 IEEE International Symposium on Hardware-Oriented Security and Trust*, 2012, pp. 89–95.
- [88] I. Bow, N. Bete, and F. Saqib, “Side-Channel Power Resistance for Encryption Algorithms Using Implementation Diversity,” *Cryptography*, vol. 4, no. 2, p. 13, 2020.
- [89] D. Das, M. Nath, B. Chatterjee, S. Ghosh, and S. Sen, “STELLAR: A Generic em Side-Channel attack protection through ground-up root-cause analysis,” *HOST 2019*, pp. 11–20, 2019, doi: 10.1109/HST.2019.8740839.
- [90] C. Boit, S. Tajik, and P. Scholz, “From IC debug to hardware security risk: The power of backside access and optical interaction,” in *2016 IEEE 23rd International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, 2016, pp. 365–369.
- [91] E. Amini, R. Muydinov, B. Szyszka, and C. Boit, “Backside protection structure for security sensitive ics,” in *Proceedings from the 43rd international symposium for testing and failure analysis*, 2017, pp. 279–284.
- [92] S. Tajik, H. Lohrke, J.-P. Seifert, and C. Boit, “On the power of optical contactless probing: Attacking bitstream encryption of FPGAs,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1661–1674.

- [93] N. Miura, D. Fujimoto, M. Nagata, N. Homma, Y. Hayashi, and T. Aoki, “EM attack sensor: concept, circuit, and design-automation methodology,” in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2015, pp. 1–6.
- [94] N. Homma, Y. I. Hayashi, and N. Miura, “Em attack is non-invasive? - Design methodology and validity verification of EM attack sensor,” *CHES 2014*, vol. 8731, pp. 1–16, 2014, doi: 10.1007/978-3-662-44709-3_1.
- [95] M. Randolph and W. Diehl, “Power Side-Channel Attack Analysis: A Review of 20 Years of Study for the Layman,” *Cryptography*, vol. 4, no. 2, p. 15, 2020.
- [96] N. Mentens, B. Gierlichs, and I. Verbauwhede, “Power and fault analysis resistance in hardware through dynamic reconfiguration,” *CHES 2008*, vol. 5154 LNCS, pp. 346–362, 2008, doi: 10.1007/978-3-540-85053-3_22.
- [97] D. Harris and S. Harris, *Digital design and computer architecture*. Morgan Kaufmann, 2010.
- [98] M. Balch, *Complete digital design: a comprehensive guide to digital electronics and computer system architecture*. McGraw-Hill Education, 2003.
- [99] L. Benini, E. Omerbegovic, A. Macii, M. Poncino, E. Macii, and F. Pro, “Energy-aware design techniques for differential power analysis protection,” in *Proceedings 2003. Design Automation Conference (IEEE Cat. No. 03CH37451)*, 2003, pp. 36–41.
- [100] M. Stöttinger, S. Malipatlolla, and Q. Tian, “Survey of methods to improve side-

- channel resistance on partial reconfigurable platforms,” in *Design Methodologies for Secure Embedded Systems*, Springer, 2010, pp. 63–84.
- [101] M. J. Herring and K. D. Willett, “Active Cyber Defense: A Vision for Real-Time Cyber Defense,” *J. Inf. Warf.*, vol. 13, no. 2014, pp. 46–55, 2014.
- [102] C. C. Davidson, “Applying Moving Target Defensive Techniques towards the Security of Programmable Logic Controllers,” University of South Alabama, 2018.
- [103] D. Evans, A. Nguyen-Tuong, and J. Knight, “Effectiveness of Moving Target Defenses,” pp. 29–48, 2011, doi: 10.1007/978-1-4614-0977-9_2.
- [104] J. Yackoski, J. Li, S. A. DeLoach, and X. Ou, “Mission-oriented moving target defense based on cryptographically strong network dynamics,” in *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, 2013, pp. 1–4.
- [105] R. Zhuang, S. A. DeLoach, and X. Ou, “A model for analyzing the effect of moving target defenses on enterprise networks,” in *Proceedings of the 9th annual cyber and information security research conference*, 2014, pp. 73–76.
- [106] K. A. Repik, “Defeating adversary network intelligence efforts with active cyber defense techniques,” 2008.
- [107] M. Atighetchi, P. Pal, F. Webber, and C. Jones, “Adaptive use of network-centric mechanisms in cyber-defense,” in *Sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, 2003.*, 2003, pp. 183–192.

- [108] F. Webber, P. P. Pal, M. Atighetchi, C. Jones, and P. Rubel, “Applications that participate in their own defense (apod),” 2003.
- [109] D. Evans, A. Nguyen-Tuong, and J. Knight, “Effectiveness of moving target defenses,” in *Moving Target Defense*, Springer, 2011, pp. 29–48.
- [110] S. Bhatkar and R. Sekar, “Data space randomization,” in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2008, pp. 1–22.
- [111] K. Sinha, V. P. Kemerlis, and S. Sethumadhavan, “Reviving instruction set randomization,” in *2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2017, pp. 21–28.
- [112] M. Azab and M. Eltoweissy, “Migrate: Towards a lightweight moving-target defense against cloud side-channels,” in *2016 IEEE Security and Privacy Workshops (SPW)*, 2016, pp. 96–103.
- [113] C. Dai and T. Adegbiya, “CONDENSE: A Moving Target Defense Approach for Mitigating Cache Side-Channel Attacks,” *IEEE Consum. Electron. Mag.*, vol. 9, no. 3, pp. 114–120, 2020.
- [114] S. Vuppala, A. E.-D. Mady, and A. Kuenzi, “Rekeying-based Moving Target Defence Mechanism for Side-Channel Attacks,” in *2019 Global IoT Summit (GIoTS)*, 2019, pp. 1–5.
- [115] G. lin Cai, B. sheng Wang, W. Hu, and T. zuo Wang, “Moving target defense:

- state of the art and characteristics,” *Front. Inf. Technol. Electron. Eng.*, vol. 17, no. 11, pp. 1122–1153, 2016, doi: 10.1631/FITEE.1601321.
- [116] T. Moos, A. Moradi, T. Schneider, and F.-X. Standaert, “Glitch-resistant masking revisited,” *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, pp. 256–292, 2019.
- [117] J.-S. Coron, “Higher order masking of look-up tables,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2014, pp. 441–458.
- [118] C. Dobraunig, M. Eichlseder, T. Korak, and F. Mendel, “Side-Channel Analysis of Keymill,” *COSADE 2017*, vol. 2, pp. 120–137, 2017, doi: 10.1007/978-3-319-64647-3.
- [119] M. Medwed, F.-X. Standaert, J. Großschädl, and F. Regazzoni, “Fresh re-keying: Security against side-channel and fault attacks for low-cost devices,” in *International Conference on Cryptology in Africa*, 2010, pp. 279–296.
- [120] X. Xi, A. Aysu, and M. Orshansky, “Fresh re-keying with strong PUFs: A new approach to side-channel security,” *HOST 2018*, pp. 118–125, 2018, doi: 10.1109/HST.2018.8383899.
- [121] S. Vuppala, “Moving Target Defense Mechanism for Side-Channel Attacks,” *IEEE Syst. J.*, pp. 1–10, 2019, doi: 10.1109/JSYST.2019.2922589.
- [122] D. May, H. L. Muller, and N. P. Smart, “Random register renaming to foil DPA,” *CHES 2001*, vol. 2162, pp. 28–38, 2001, doi: 10.1007/3-540-44709-1_4.

- [123] K. Itoh, T. Izu, and M. Takenaka, “A practical countermeasure against address-bit differential power analysis,” *CHES 2003*, vol. 2779, pp. 382–396, 2003, doi: 10.1007/978-3-540-45238-6_30.
- [124] H. Mamiya, A. Miyaji, and H. Morimoto, “Efficient countermeasures against RPA, DPA, and SPA,” *CHES 2004*, vol. 3156, pp. 343–356, 2004, doi: 10.1007/978-3-540-28632-5_25.
- [125] J.-S. Coron, “Resistance against differential power analysis for elliptic curve cryptosystems,” *CHES 1999*, vol. 1717, pp. 292–302, 1999, doi: 10.1007/3-540-48059-5_25.
- [126] M. Ciet and M. Joye, “(Virtually) free randomization techniques for elliptic curve cryptography,” in *International Conference on Information and Communications Security*, 2003, pp. 348–359.
- [127] Z. Liu, P. Longa, G. Pereira, O. Reparaz, and H. Seo, “FourQ on embedded devices with strong countermeasures against side-channel attacks,” *CHES 2017*, pp. 1–21, 2017, doi: 10.1109/TDSC.2018.2799844.
- [128] Z. Zhang, Q. Yu, L. Njilla, and C. Kamhoua, “FPGA-oriented moving target defense against security threats from malicious FPGA tools,” *Proc. 2018 IEEE Int. Symp. Hardw. Oriented Secur. Trust. HOST 2018*, pp. 163–166, 2018, doi: 10.1109/HST.2018.8383907.
- [129] Z. Zhang, L. Njilla, C. A. Kamhoua, and Q. Yu, “Thwarting security threats from

- malicious FPGA tools with Novel FPGA-Oriented moving target defense,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 27, no. 3, pp. 665–678, 2019, doi: 10.1109/TVLSI.2018.2879878.
- [130] B. Blodget, C. Bobda, M. Huebner, and A. Niyonkuru, “Partial and dynamically reconfiguration of Xilinx Virtex-II FPGAs,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3203, pp. 801–810, 2004, doi: 10.1007/978-3-540-30117-2_81.
- [131] M. Stöttinger, F. Madlener, and S. A. Huss, “Procedures for securing ECC implementations against differential power analysis using reconfigurable architectures,” in *Dynamically Reconfigurable Systems: Architectures, Design Methods and Applications*, Springer, 2010, pp. 395–415.
- [132] G. Bloom, B. Narahari, R. Simha, A. Namazi, and R. Levy, “FPGA SoC architecture and runtime to prevent hardware Trojans from leaking secrets,” *Proc. 2015 IEEE Int. Symp. Hardware-Oriented Secur. Trust. HOST 2015*, pp. 48–51, 2015, doi: 10.1109/HST.2015.7140235.
- [133] S. A. Huss and M. Stöttinger, “A novel mutating runtime architecture for embedding multiple countermeasures against side-channel attacks,” in *Hardware IP security and trust*, Springer, 2017, pp. 165–184.
- [134] M. A. Forbes, *Digital Logic Protection Using Functional Polymorphism and Topology Hiding*. University of South Alabama, 2017.

- [135] J. T. McDonald, D. McKinney, and T. R. Andel, “Program Encryption Toolkit: A Tool for Digital Logic Education and Undergraduate Research,” 2021.
- [136] J. T. McDonald, Y. C. Kim, and M. R. Grimaila, “Protecting reprogrammable hardware with polymorphic circuit variation,” in *Proceedings of the 2nd Cyberspace Research Workshop*, 2009, pp. 63–78.
- [137] Xilinx, *Vivado Design Suite User Guide: Synthesis*. Xilinx, 2020.
- [138] Xilinx, *Vivado Design Suite User Guide: Implementation*. Xilinx, 2021.
- [139] Xilinx, “Zynq-7000 SoC product selection guide,” *Xilinx*. 2019, [Online]. Available: <https://www.xilinx.com/support/documentation/selection-guides/zynq-7000-product-selection-guide.pdf>.
- [140] “ZedBoard Hardware User Guide,” *Xilinx*. [Online]. Available: http://zedboard.org/sites/default/files/documentations/ZedBoard_HW_UG_v2_2.pdf.

APPENDICES

Appendix A: First Order Analysis of Control Implementation

Results after 291 traces

Best score for Round 0: Key: Column 0, Row 0 with rdm: 0.0670:
rank: 1, candidate: 35 (0x23), confidence: 0.2981 at position: 48062
rank: 2, candidate: 68 (0x44), confidence: 0.2966 at position: 30644
rank: 3, candidate: 251 (0xFB), confidence: 0.2879 at position: 18634
rank: 4, candidate: 235 (0xEB), confidence: 0.2852 at position: 43777
Best score for Round 0: Key: Column 0, Row 1 with rdm: 1.9437:
rank: 1, candidate: 20 (0x14), confidence: 0.3463 at position: 20580
rank: 2, candidate: 43 (0x2B), confidence: 0.3008 at position: 51257
rank: 3, candidate: 225 (0xE1), confidence: 0.2978 at position: 16956
rank: 4, candidate: 247 (0xF7), confidence: 0.2957 at position: 28802
Best score for Round 0: Key: Column 0, Row 2 with rdm: 0.1265:
rank: 1, candidate: 190 (0xBE), confidence: 0.2940 at position: 19466
rank: 2, candidate: 221 (0xDD), confidence: 0.2912 at position: 29566
rank: 3, candidate: 114 (0x72), confidence: 0.2908 at position: 15700
rank: 4, candidate: 91 (0x5B), confidence: 0.2908 at position: 36400
Best score for Round 0: Key: Column 0, Row 3 with rdm: 0.5277:
rank: 1, candidate: 80 (0x50), confidence: 0.3176 at position: 39715
rank: 2, candidate: 94 (0x5E), confidence: 0.3060 at position: 29584
rank: 3, candidate: 128 (0x80), confidence: 0.2988 at position: 37054
rank: 4, candidate: 87 (0x57), confidence: 0.2856 at position: 22344
Best score for Round 0: Key: Column 1, Row 0 with rdm: 0.6389:
rank: 1, candidate: 246 (0xF6), confidence: 0.3046 at position: 25695
rank: 2, candidate: 127 (0x7F), confidence: 0.2914 at position: 38434
rank: 3, candidate: 202 (0xCA), confidence: 0.2857 at position: 51898
rank: 4, candidate: 163 (0xA3), confidence: 0.2802 at position: 29464
Best score for Round 0: Key: Column 1, Row 1 with rdm: 0.6371:
rank: 1, candidate: 39 (0x27), confidence: 0.3128 at position: 44318
rank: 2, candidate: 14 (0x0E), confidence: 0.2987 at position: 18463
rank: 3, candidate: 83 (0x53), confidence: 0.2850 at position: 22103
rank: 4, candidate: 222 (0xDE), confidence: 0.2837 at position: 42815
Best score for Round 0: Key: Column 1, Row 2 with rdm: 0.3041:
rank: 1, candidate: 94 (0x5E), confidence: 0.2934 at position: 26035
rank: 2, candidate: 249 (0xF9), confidence: 0.2867 at position: 6117
rank: 3, candidate: 209 (0xD1), confidence: 0.2798 at position: 26610
rank: 4, candidate: 62 (0x3E), confidence: 0.2795 at position: 28656
Best score for Round 0: Key: Column 1, Row 3 with rdm: 0.4298:
rank: 1, candidate: 83 (0x53), confidence: 0.3052 at position: 26441
rank: 2, candidate: 56 (0x38), confidence: 0.2955 at position: 45064

```

rank: 3, candidate: 176 (0xB0), confidence: 0.2936 at position: 9926
rank: 4, candidate: 54 (0x36), confidence: 0.2902 at position: 46528
Best score for Round 0: Key: Column 2, Row 0 with rdm: 0.7739:
rank: 1, candidate: 48 (0x30), confidence: 0.3012 at position: 45614
rank: 2, candidate: 132 (0x84), confidence: 0.2840 at position: 19165
rank: 3, candidate: 251 (0xFB), confidence: 0.2840 at position: 18246
rank: 4, candidate: 157 (0x9D), confidence: 0.2837 at position: 46269
Best score for Round 0: Key: Column 2, Row 1 with rdm: 0.2525:
rank: 1, candidate: 222 (0xDE), confidence: 0.3229 at position: 33403
rank: 2, candidate: 175 (0xAF), confidence: 0.3168 at position: 44658
rank: 3, candidate: 196 (0xC4), confidence: 0.2968 at position: 6330
rank: 4, candidate: 20 (0x14), confidence: 0.2872 at position: 29339
Best score for Round 0: Key: Column 2, Row 2 with rdm: 0.2162:
rank: 1, candidate: 236 (0xEC), confidence: 0.3049 at position: 41593
rank: 2, candidate: 233 (0xE9), confidence: 0.3000 at position: 45353
rank: 3, candidate: 173 (0xAD), confidence: 0.2955 at position: 24858
rank: 4, candidate: 207 (0xCF), confidence: 0.2889 at position: 52198
Best score for Round 0: Key: Column 2, Row 3 with rdm: 0.6103:
rank: 1, candidate: 13 (0x0D), confidence: 0.3101 at position: 25167
rank: 2, candidate: 204 (0xCC), confidence: 0.2964 at position: 26499
rank: 3, candidate: 225 (0xE1), confidence: 0.2946 at position: 42044
rank: 4, candidate: 130 (0x82), confidence: 0.2937 at position: 31988
Best score for Round 0: Key: Column 3, Row 0 with rdm: 0.8589:
rank: 1, candidate: 215 (0xD7), confidence: 0.3271 at position: 6947
rank: 2, candidate: 169 (0xA9), confidence: 0.3086 at position: 11477
rank: 3, candidate: 224 (0xE0), confidence: 0.2990 at position: 34315
rank: 4, candidate: 99 (0x63), confidence: 0.2868 at position: 37140
Best score for Round 0: Key: Column 3, Row 1 with rdm: 0.2344:
rank: 1, candidate: 46 (0x2E), confidence: 0.2985 at position: 25770
rank: 2, candidate: 29 (0x1D), confidence: 0.2938 at position: 35983
rank: 3, candidate: 45 (0x2D), confidence: 0.2776 at position: 41323
rank: 4, candidate: 31 (0x1F), confidence: 0.2775 at position: 38632
Best score for Round 0: Key: Column 3, Row 2 with rdm: 0.0231:
rank: 1, candidate: 250 (0xFA), confidence: 0.2970 at position: 50299
rank: 2, candidate: 56 (0x38), confidence: 0.2964 at position: 37791
rank: 3, candidate: 158 (0x9E), confidence: 0.2952 at position: 7780
rank: 4, candidate: 61 (0x3D), confidence: 0.2930 at position: 43594
Best score for Round 0: Key: Column 3, Row 3 with rdm: 0.6883:
rank: 1, candidate: 128 (0x80), confidence: 0.3111 at position: 48872
rank: 2, candidate: 60 (0x3C), confidence: 0.2953 at position: 32478
rank: 3, candidate: 226 (0xE2), confidence: 0.2934 at position: 48927
rank: 4, candidate: 85 (0x55), confidence: 0.2838 at position: 13568
Unverified key:
00100011000101001011111001010000111101100010011101011110010100110011000
011011110111011000000110111010111001011101111101010000000/0 bits
entropy remain (0x2314be50f6275e5330deec0dd72efa80)
Key can not be verified. Setting key to the most likely value
Detailed key info:
00100011000101001011111001010000111101100010011101011110010100110011000
011011110111011000000110111010111001011101111101010000000/0 bits
entropy remain (0x2314be50f6275e5330deec0dd72efa80)

```

Appendix B: First Order Analysis of Countermeasure Implementation

Results after 1997 traces

Best score for Round 0: Key: Column 0, Row 0 with rdm: 0.8468:
rank: 1, candidate: 240 (0xF0), confidence: 0.1192 at position: 10937
rank: 2, candidate: 168 (0xA8), confidence: 0.1130 at position: 7708
rank: 3, candidate: 241 (0xF1), confidence: 0.1100 at position: 7782
rank: 4, candidate: 200 (0xC8), confidence: 0.1096 at position: 11158
Best score for Round 0: Key: Column 0, Row 1 with rdm: 0.8235:
rank: 1, candidate: 137 (0x89), confidence: 0.1192 at position: 4780
rank: 2, candidate: 210 (0xD2), confidence: 0.1134 at position: 9034
rank: 3, candidate: 162 (0xA2), confidence: 0.1084 at position: 8511
rank: 4, candidate: 37 (0x25), confidence: 0.1059 at position: 3141
Best score for Round 0: Key: Column 0, Row 2 with rdm: 0.5825:
rank: 1, candidate: 254 (0xFE), confidence: 0.1075 at position: 4349
rank: 2, candidate: 202 (0xCA), confidence: 0.1037 at position: 8938
rank: 3, candidate: 220 (0xDC), confidence: 0.1030 at position: 3767
rank: 4, candidate: 122 (0x7A), confidence: 0.1029 at position: 4382
Best score for Round 0: Key: Column 0, Row 3 with rdm: 0.0877:
rank: 1, candidate: 217 (0xD9), confidence: 0.1104 at position: 4346
rank: 2, candidate: 51 (0x33), confidence: 0.1098 at position: 4479
rank: 3, candidate: 161 (0xA1), confidence: 0.1070 at position: 7542
rank: 4, candidate: 198 (0xC6), confidence: 0.1063 at position: 6665
Best score for Round 0: Key: Column 1, Row 0 with rdm: 0.8138:
rank: 1, candidate: 221 (0xDD), confidence: 0.1181 at position: 8733
rank: 2, candidate: 155 (0x9B), confidence: 0.1123 at position: 10696
rank: 3, candidate: 234 (0xEA), confidence: 0.1064 at position: 9562
rank: 4, candidate: 63 (0x3F), confidence: 0.1057 at position: 10620
Best score for Round 0: Key: Column 1, Row 1 with rdm: 0.5135:
rank: 1, candidate: 228 (0xE4), confidence: 0.1096 at position: 6555
rank: 2, candidate: 232 (0xE8), confidence: 0.1061 at position: 8505
rank: 3, candidate: 105 (0x69), confidence: 0.1058 at position: 7217
rank: 4, candidate: 123 (0x7B), confidence: 0.1045 at position: 9677
Best score for Round 0: Key: Column 1, Row 2 with rdm: 0.7374:
rank: 1, candidate: 228 (0xE4), confidence: 0.1147 at position: 5587
rank: 2, candidate: 184 (0xB8), confidence: 0.1097 at position: 10518
rank: 3, candidate: 30 (0x1E), confidence: 0.1041 at position: 6901
rank: 4, candidate: 227 (0xE3), confidence: 0.1039 at position: 10733
Best score for Round 0: Key: Column 1, Row 3 with rdm: 0.6371:
rank: 1, candidate: 38 (0x26), confidence: 0.1146 at position: 3999
rank: 2, candidate: 106 (0x6A), confidence: 0.1102 at position: 8883
rank: 3, candidate: 79 (0x4F), confidence: 0.1085 at position: 9064
rank: 4, candidate: 154 (0x9A), confidence: 0.1056 at position: 3795
Best score for Round 0: Key: Column 2, Row 0 with rdm: 0.3561:
rank: 1, candidate: 118 (0x76), confidence: 0.1118 at position: 5182
rank: 2, candidate: 155 (0x9B), confidence: 0.1094 at position: 8607
rank: 3, candidate: 146 (0x92), confidence: 0.1078 at position: 7069
rank: 4, candidate: 64 (0x40), confidence: 0.1073 at position: 6394
Best score for Round 0: Key: Column 2, Row 1 with rdm: 0.3088:
rank: 1, candidate: 73 (0x49), confidence: 0.1149 at position: 4042
rank: 2, candidate: 178 (0xB2), confidence: 0.1126 at position: 3860
rank: 3, candidate: 54 (0x36), confidence: 0.1120 at position: 9870
rank: 4, candidate: 74 (0x4A), confidence: 0.1096 at position: 11223
Best score for Round 0: Key: Column 2, Row 2 with rdm: 0.1465:

```

rank: 1, candidate: 30 (0x1E), confidence: 0.1116 at position: 6058
rank: 2, candidate: 204 (0xCC), confidence: 0.1105 at position: 9682
rank: 3, candidate: 91 (0x5B), confidence: 0.1103 at position: 10178
rank: 4, candidate: 166 (0xA6), confidence: 0.1097 at position: 6207
Best score for Round 0: Key: Column 2, Row 3 with rdm: 1.3435:
rank: 1, candidate: 150 (0x96), confidence: 0.1192 at position: 5444
rank: 2, candidate: 211 (0xD3), confidence: 0.1100 at position: 3800
rank: 3, candidate: 227 (0xE3), confidence: 0.1056 at position: 4846
rank: 4, candidate: 191 (0xBF), confidence: 0.1035 at position: 5833
Best score for Round 0: Key: Column 3, Row 0 with rdm: 0.1105:
rank: 1, candidate: 249 (0xF9), confidence: 0.1090 at position: 9034
rank: 2, candidate: 88 (0x58), confidence: 0.1082 at position: 7995
rank: 3, candidate: 165 (0xA5), confidence: 0.1060 at position: 4871
rank: 4, candidate: 197 (0xC5), confidence: 0.1048 at position: 8452
Best score for Round 0: Key: Column 3, Row 1 with rdm: 0.5972:
rank: 1, candidate: 50 (0x32), confidence: 0.1125 at position: 5765
rank: 2, candidate: 138 (0x8A), confidence: 0.1087 at position: 10125
rank: 3, candidate: 109 (0x6D), confidence: 0.1075 at position: 3557
rank: 4, candidate: 123 (0x7B), confidence: 0.1050 at position: 9636
Best score for Round 0: Key: Column 3, Row 2 with rdm: 0.3551:
rank: 1, candidate: 72 (0x48), confidence: 0.1132 at position: 3154
rank: 2, candidate: 213 (0xD5), confidence: 0.1107 at position: 3490
rank: 3, candidate: 50 (0x32), confidence: 0.1086 at position: 9217
rank: 4, candidate: 144 (0x90), confidence: 0.1071 at position: 8500
Best score for Round 0: Key: Column 3, Row 3 with rdm: 0.0059:
rank: 1, candidate: 28 (0x1C), confidence: 0.1213 at position: 5891
rank: 2, candidate: 64 (0x40), confidence: 0.1212 at position: 5422
rank: 3, candidate: 70 (0x46), confidence: 0.1121 at position: 3698
rank: 4, candidate: 249 (0xF9), confidence: 0.1095 at position: 10418
Unverified key:
11110000100010011111111011011001110111011110010011100100001001100111011
001001001000111101001011011111001001100100100100000011100/0 bits
entropy remain (0xf089fed9dde4e42676491e96f932481c)
Key can not be verified. Setting key to the most likely value
Detailed key info:
11110000100010011111111011011001110111011110010011100100001001100111011
001001001000111101001011011111001001100100100100000011100/0 bits
entropy remain (0xf089fed9dde4e42676491e96f932481c)

```

BIOGRAPHICAL SKETCH

Name of Author: Tristen H. Mullins

Graduate and Undergraduate Schools Attended:

University of South Alabama, Mobile, Alabama

Degrees Awarded:

Doctor of Philosophy in Computing, 2022, Mobile, Alabama

Bachelor of Science in Computer Engineering, 2018, Mobile, Alabama

Awards and Honors:

National Science Foundation CyberCorps Scholarship For Service Student, 2019-2022

Publications:

Tristen H. Mullins, Todd R. Andel, and J. Todd McDonald. Circuit Variant Moving Target Defense for Side-Channel Attacks. 17th International Conference on Cyber Warfare and Security, New York, USA, pages 219-226, 2022.